
Fachbereich Automatisierung und Informatik
Studiengang Intelligente Automatisierungssysteme
Eingereicht Februar 2012

Bachelorarbeit

Entwicklung eines Systems zur Ortung von Straßen- und Schienenfahrzeugen vom Entwurf bis zur Einsatzreife

eingereicht von
Georg Richter

Matrikelnummer
14784

Erstprüfer: **Prof. Dr. rer. nat. habil. Frieder Stolzenburg**
Hochschule Harz Wernigerode
Fachbereich Automatisierung und Informatik

Zweitprüfer: **Dipl.-Inf. (FH) Andreas Kluge**
Deutsches Zentrum für Luft- und Raumfahrt Braunschweig
Institut für Verkehrssystemtechnik

Inhaltsverzeichnis

1. Einleitung	1
1.1. Einordnung	1
1.2. Aufgaben- und Zielstellung	2
2. Grundlagen	3
2.1. Grundlagen der Ortung und Navigation	3
2.2. Erhöhung der Genauigkeit	5
2.2.1. Die Unterscheidung nach der Anzahl an Datenquellen	5
2.2.2. Systeme mit nur einer Datenquelle	7
2.2.3. Systeme mit mehreren Datenquellen	7
2.2.4. Filterverfahren und Datenfusion	8
2.2.4.1. Vorbemerkungen	8
2.2.4.2. Der Wiener-Filter	8
2.2.4.3. Der Kalman-Filter	9
2.2.4.4. Markow-Modelle	11
2.2.4.5. Map-Matching	12
2.2.4.6. SLAM	15
2.3. Ortungssysteme für Fahrzeuge	17
2.4. Kommunikation und Datenfunk im ÖPNV	20
2.4.1. Analogfunksysteme	20
2.4.2. Digitalfunksysteme	20
2.4.2.1. Überblick	20
2.4.2.2. DMR	21
2.4.2.3. dPMR	21
2.4.2.4. TETRA	22
2.4.2.5. TETRAPOL	23
2.4.2.6. GSM	24
2.5. Konzept des Ortungsmoduls	26
2.5.1. Grundanforderungen und Einsatzszenario	26
2.5.2. Prinzipieller Aufbau	27
2.6. Bestehende vergleichbare Systeme	29

3. Hardware	30
3.1. Anforderungen, Auswahlkriterien und -verfahren	30
3.2. Auswahl des Mikrocontrollers	31
3.3. Auswahl des GPS-Moduls	32
3.3.1. Vorbemerkung	32
3.3.2. Aufbau eines GPS-Empfängers	32
3.3.3. Arbeitsweise eines GPS-Empfängers	34
3.3.4. Verfügbarkeit und Auswahl der GPS-Module	36
3.3.5. Evaluation ausgewählter GPS-Module	36
3.3.5.1. Überblick	36
3.3.5.2. Vergleich der Spezifikationen laut Hersteller	37
3.3.5.3. Messung der TTFF und Updaterate	38
3.3.5.4. Vergleich der Ortungsgenauigkeit im stationären Einsatz	39
3.3.5.5. Vergleich der Ortungsgenauigkeit in Bewegung	43
3.4. Auswahl des GSM-Moduls	47
3.5. Zusatzmodule und Zubehör	48
3.6. Der erste Prototyp	49
3.6.1. Aufbau und Entwicklungsgeschichte	49
3.6.2. Endstand	51
4. Software	53
4.1. Entwicklungskonzept	53
4.2. Gegebene Softwarekomponenten	55
4.2.1. Die Arduino-Umgebung	55
4.2.2. GPS-Bibliothek	56
4.2.3. GSM/GPRS-Bibliothek	56
4.3. Entwicklung erster Testprogramme	57
4.3.1. GPS-Benutzung	57
4.3.2. SD-Logging	57
4.3.3. GSM- und GPRS-funktionen	58
4.3.3.1. GSM-Telefoniefunktionen	58
4.3.3.2. GPRS-Datenkommunikation	58
4.4. Firmware des Gerätes: Anforderungen und geplanter Funktionsumfang	59
4.4.1. Ortung, Logging und Datenaufbereitung	59
4.4.2. Konfiguration	59
4.4.3. Kommunikation und Datenübertragung	60
4.5. Die AIM-Webservices als Datenempfänger	60
4.6. Entwicklung der Gerätefirmware	61
4.6.1. Konzeption	61
4.6.1.1. Überblick	61

4.6.1.2.	Kommunikation zwischen beteiligten Komponenten	62
4.6.1.3.	Datenverarbeitung im Mikrocontroller	62
4.6.1.4.	Abstraktion der Hardware durch Klassen	64
4.6.2.	Implementierung	65
4.6.2.1.	Struktur	65
4.6.2.2.	Herkunft der Softwarebestandteile	67
4.6.2.3.	Eigener Code	68
4.6.2.4.	Modifizierung fremder Bibliotheken	70
4.6.3.	Nutzung	71
4.6.3.1.	Erstellung und Aufspielen auf den Controller	71
4.6.3.2.	Konfiguration	72
5.	Ergebnisse und deren Bewertung	74
5.1.	Hardware	74
5.1.1.	Derzeitiger Stand / erreichter Funktionsumfang	74
5.1.2.	Vorhandene Probleme	74
5.1.3.	Bewertung der Einsatztauglichkeit	75
5.2.	Software	76
5.2.1.	Derzeitiger Stand / erreichter Funktionsumfang	76
5.2.2.	Vorhandene Probleme	76
5.2.3.	Bewertung der Einsatzreife und Serientauglichkeit	77
5.3.	Funktion	77
5.3.1.	Qualität und Aktualität übertragener Informationen	77
5.3.2.	Zuverlässigkeit des Gesamtsystems im Realeinsatz	78
6.	Zusammenfassung	80
7.	Ausblick	81
7.1.	Laufende und geplante Weiterentwicklung des Gerätes	81
7.1.1.	Hardware	81
7.1.1.1.	Ablösung des Ebenen-Systems	81
7.1.1.2.	Erreichen der Serienreife	82
7.1.2.	Software	83
7.2.	Anwendungsmöglichkeiten der nun verfügbaren Informationen	83
	Abbildungsverzeichnis	85
	Tabellenverzeichnis	86
	Quellenverzeichnis	87
	Literaturverzeichnis	91

Abkürzungsverzeichnis	93
A. Vergleich verschiedener Hardwarekomponenten	A-1
B. Messprotokolle zum Vergleich ausgewählter GPS-Module	B-1
B.1. Anmerkungen	B-1
B.2. Updateraten	B-1
B.3. Startzeiten	B-2
C. Schaltpläne	C-1
C.1. Pegelanpassung des Venus-GPS-Moduls	C-1
D. Sourcecodes und technische Unterlagen	D-1

1. Einleitung

1.1. Einordnung

Der Personen- und Materialtransport auf Straße und Schiene ist treibender Motor unserer Wirtschaft. Die Optimierung des Verkehrsflusses, die Verbesserung der Interoperabilität verschiedener Verkehrsträger sowie die Bewältigung steigender Verkehrslasten sind zentrale Aufgaben der heutigen Infrastrukturpolitik. Innovative Ansätze zur Nutzung moderner Technologien tragen entscheidend dazu bei, aktuellen und zukünftigen Herausforderungen in der Verkehrsgestaltung zu begegnen.

Das Institut für Verkehrssystemtechnik des Deutschen Zentrums für Luft- und Raumfahrt e.V. (DLR) schafft zusammen mit weiteren Partnern mit der *Anwendungsplattform Intelligente Mobilität* (AIM) eine bisher einzigartige Möglichkeit zur Erforschung, Entwicklung und Erprobung moderner Transport- und Mobilitätsdienste. Neben Simulationsansätzen und kontrollierten Testumgebungen setzt das Projekt insbesondere auf die Nutzung des Verkehrsraumes Braunschweig als Modellregion zur Gewinnung realer Verkehrsdaten sowie zur Evaluierung neuartiger Technologiekonzepte.

Das Wissen, welches die zentrale Komponente dieser Plattform darstellt, gründet sich auf Informationen und Daten, die von zahlreichen im Rahmen dieses Großforschungsprojektes angelegten Teilprojekten erfasst und zur Verfügung gestellt werden. Eines dieser Teilprojekte widmet sich der Bewegung individueller Fahrzeuge. Besonderes Interesse gilt hierbei dem öffentlichen Personennahverkehr, da dessen mobile Einheiten wie Busse und Straßenbahnen besonders günstige Bedingungen zur Erfassung verschiedenster Verkehrssituationen bieten, darunter die regelmäßige Nutzung wiederkehrender Routen zu unterschiedlichen Uhrzeiten sowie die langfristige Verfügbarkeit einer beständigen Fahrzeugflotte zur Ausstattung mit hochwertiger Sensorik. Die Entwicklung einer Komponente zur langfristigen und kontinuierlichen Erfassung der Position und Bewegung einzelner Fahrzeuge ist Inhalt dieser Arbeit.

1.2. Aufgaben- und Zielstellung

Straßen- und Schienenfahrzeuge legen täglich unzählige Kilometer zurück. Dabei wird es immer wichtiger, exakte Informationen über den aktuellen Ort und den Verlauf der Bewegung eines Objektes zur Verfügung zu haben. Terminlich exakte Lieferungen sind heute ebenso essenzielle Bestandteile unserer Wirtschaft wie die Wegverfolgung wertvollen Frachtgutes. Im öffentlichen Nah- und Fernverkehr sind Informationen über Verkehrsaufkommen und Störungen Grundvoraussetzung für eine Optimierung des Betriebsablaufes sowie die Einhaltung von Fahrplänen. Moderne Anwendungen zur Stauvorhersage, -vermeidung und Navigation benötigen eine hochaktuelle, präzise Datenbasis für ihre Entscheidungen.

Ziel der Bachelorarbeit ist die Entwicklung eines kostengünstigen, universell einsetzbaren Systems, welches seine per GPS erfassten Positionsdaten kontinuierlich und in Echtzeit mittels GSM an einen zentralen Webservice übertragen kann. Das fertige System soll für den Betrieb in Bussen und Straßenbahnen ebenso geeignet sein wie für Eisenbahnen, LKW, Taxen und autarkes Frachtgut. Eine spätere Kleinserienerprobung im realen Einsatz soll vorgesehen werden.

Die Bachelorarbeit beinhaltet folgende Teilaufgaben:

- Analyse vorhandener Lösungen für GPS-Ortung und GSM-Datenübertragung
- Bau und Evaluierung eines Prototypen aus vorhandenen Komponenten
- Entwurf eines eigenen Gerätes mit besonderem Augenmerk auf universelle Einsetzbarkeit, Kostenreduzierung und Entwicklung bis zur Anwendungsreife
- Dokumentation und Bewertung des Projektes unter Anwendung softwaretechnischer Methoden

2. Grundlagen

2.1. Grundlagen der Ortung und Navigation

Der Begriff der *Ortung* beschreibt die Bestimmung einer Position gegenüber einem Bezugssystem. Die *Navigation* bezeichnet streng genommen die Verwendung von Positionsinformationen zur Orientierung gegenüber eines in einem solchen Bezugssystem definierten Zieles oder Referenzpunktes. Beide Begriffe sind jedoch sehr eng miteinander verbunden und werden häufig austauschbar verwendet. Die im Rahmen dieser Arbeit auftretenden Größenordnungen meinen zumeist das Bestimmen und Kennen der eigenen Position mit einer Genauigkeit im Bereich von Metern. Das Ziel ist es, die Zuordnung des eigenen Aufenthaltsortes etwa zu einer bestimmten Straße oder Kreuzung zu ermöglichen, das Bezugssystem sind globale Koordinatensysteme wie etwa das international übliche *World Geodatic System 1984* (WGS84).

Nach [45, S. 1] unterscheidet man fünf wesentliche Formen der Navigation:

- Lotsung
- Koppelnavigation
- Astronomische Navigation
- Radionavigation
- Trägheitsnavigation

Die *Lotsung* (oder auch Führung i.S.v. engl. *pilotage* oder *guidance*) basiert auf der (Wieder-)Erkennung markanter Punkte der Landschaft entlang eines geplanten Weges. Sie ist die klassische Form der Orientierung insbesondere sehender Lebewesen und damit weitaus älter als die Menschheit. Sie kommt für technologische Anwendungen bisher prinzipbedingt erst wenig in Frage, da zur Bestimmung einer Position stets aktuelle Referenzdaten über geeignete markante Punkte vorliegen müssen, welche je nach Maßstab enorm variieren (einzelne Bäume oder Gebäude zur Orientierung für Fußgänger, aber Berglandschaften oder Flussverläufe für die Luftfahrt). Zudem müssen diese Referenzpunkte unter verschiedensten Sicht- und Witterungsbedingungen aus allen erdenklichen Perspektiven erkannt werden können – eine Aufgabe, die künstliche Systeme außerhalb speziell

für sie geschaffener Räume bisher nicht annähernd in der geforderten Geschwindigkeit und Präzision leisten können.

Koppelnavigation bezeichnet die Bestimmung eines Ortes durch permanente Erfassung der Geschwindigkeit, Zeit und Bewegungsrichtung eines Objektes. Klassisch geschieht dies mittels einer hinreichend präzisen Uhr, einem Fahrtmesser und einem Kompass. Für technische Anwendungen wird der Fahrtmesser oft durch odometrische Verfahren ersetzt (Weg / Zeit ergibt Geschwindigkeit). Durch *koppeln* dieser Daten lässt sich die vollführte Bewegung von einem bekannten Startpunkt aus zurückverfolgen, was die aktuelle Position ergibt. Dieses Verfahren ist prinzipiell ebenfalls sehr lange bekannt und wird auch technologisch häufig genutzt. Seine größte Unzulänglichkeit besteht darin, dass die Positionsinformation durch unvermeidbare Ungenauigkeiten in der Erfassung der Einzelparameter mit fortdauernder Nutzung einem immer größer werdenden Fehler unterliegt, weswegen ein zusätzliches System zur regelmäßigen Synchronisation (Nullung) benötigt wird.

Die *astronomische Navigation* basiert auf der Peilung bekannter astronomischer Objekte, zumeist der Sonne, des Mondes und fremder Sterne. Deren Winkel (Himmelsrichtung und Höhe über dem Horizont) zu gegebenen Zeitpunkten ermöglichen die Abbildung der eigenen Position in ein Koordinatensystem. Dieses System wird von Menschen vor allem in der Seefahrt seit Jahrhunderten genutzt, wovon sich die bis heute verwendete Angabe globaler Positionen in Längen- und Breitengraden ableitet. Heute kommt die Astronavigation auch in der Raumfahrt zum Einsatz, wo weit außerhalb der Erde keine anderen Bezugssysteme mehr vorhanden sind. Die Genauigkeit dieses Ansatzes liegt jedoch in Maßstäben von vielen Kilometern bis hin zu astronomischen Einheiten, er ist kaum für die lokale Orientierung an Land geeignet.

Radionavigationssysteme nutzen Quellen elektromagnetischer Strahlung mit bekannter Position, um über verschiedenste Effekte daraus ihre eigene Position herzuleiten. Die von den Sendern ausgehenden Signale weisen dabei definierte Eigenschaften auf und sind fast immer mit zusätzlichen Informationen angereichert, welche für die Funktion des jeweiligen Systems benötigt werden. Die Bandbreite reicht von optischen Systemen über terrestrische Radiosender (Funknavigation) bis hin zu modernen Satellitennavigationssystemen, sie bilden eine der größten Gruppen heutiger technisch unterstützter Ortungsverfahren.

Die *Trägheitsnavigation* oder auch *inertiale Navigation* basiert auf Grundeffekten der Physik. Ausgehend von einer bekannten initialen Position, Geschwindigkeit und Bewegungsrichtung werden die lokal (am bewegten Objekt) auftretenden Beschleunigungen und Drehbewegungen erfasst und aufintegriert, wodurch der zurückgelegte Weg und damit die Veränderung der eigenen Position lückenlos nachvollziehbar wird. Der entscheidende Unterschied zur Koppelnavigation liegt in der Art der erfassten Parameter: *Inertiale Navigationssysteme* (INS) sind die einzigen, welches auf keinerlei externe Referenzen (Erdmagnetfeld, sichtbare Peilungspunkte, ...) angewiesen sind und die vollständig autonom

funktionieren können. Sie arbeiten insbesondere kurz nach dem Start mit einer sehr hohen Genauigkeit, leiden ähnlich wie die Koppelnavigation jedoch an einem permanent anwachsenden Fehler mit zunehmender Dauer der Nutzung [47, Kapitel 3.4].

Heutige Anwendungen verbinden oft mehrere dieser Ansätze, um die Qualität der gewonnenen Positionsinformation zu erhöhen. Insbesondere die Unterstützung von Satellitenortungssystemen durch inertielle Navigationssysteme wird sehr häufig angewandt, da diese sehr gegenläufige und sich ergänzende Eigenschaften aufweisen. Die Fusion mehrerer Datenquellen ist eines der aktivsten Forschungsgebiete moderner Navigationsanwendungen und Gegenstand zahlreicher Veröffentlichungen ([45], [46], [47], [58]).

2.2. Erhöhung der Genauigkeit

2.2.1. Die Unterscheidung nach der Anzahl an Datenquellen

Ortungstechnologien können auf viele Weise klassifiziert werden. Unter dem Gesichtspunkt der am Ende gewonnenen Information besteht eine Möglichkeit in der Unterscheidung zwischen einfachen Systemen, welche unter Verarbeitung von Daten aus genau einer Datenquelle eine Positionsinformation liefern sowie komplexeren Systemen, welche Daten mehrerer Quellen fusionieren. Steht nur eine Datenquelle zur Verfügung, muss es sich bei dieser stets um einen Sensor handeln – ohne ein reales Eingangssignal kann ein System kein davon abhängiges Ergebnis liefern. Im zweiten Falle ist jedoch eine weitere Unterscheidung möglich: Stehen *mehrere Sensoren* zur Verfügung oder wird weiterhin nur *ein Sensor* verwendet, das System aber mit weiteren Informationen versorgt?

Verdeutlichen lässt sich dies an einem einfachen Beispiel mit Hilfe von Abb. 2.1: Ein Wanderer verlässt eine Stadt in Richtung Norden entlang des weißen Weges. Dank seines Schrittzählers weiß er stets, wie weit er gegangen ist (Wegstrecke x , blau gepunktete Linie). Mit zunehmender Entfernung kann er immer weniger genau sagen, wo er sich befindet. Er weiß nur noch, dass er sich in ungefähr nördlicher Richtung innerhalb des Radius x befinden muss, da der von ihm gegangene Weg nicht völlig gerade verläuft. Sein *einzigster Sensor* gibt nicht mehr die genaue (oder kürzeste) Entfernung zur Stadt an, aus einer exakt bestimmbaren Position wird somit ein grob umrissenes Gebiet.

Erst mit einer maßstäblichen Karte wäre es dem Wanderer möglich, den gegangenen weißen Weg um die entsprechende Entfernung zu verfolgen und daraus seine Position abzuleiten (*ein Sensor und eine Informationsquelle*). Da ein Schrittzähler jedoch für größere Entfernungen nicht beliebig genau ist, wird er nicht den (wahren) Punkt A als seinen Aufenthaltsort bestimmen, sondern ihn um eine gewisse Entfernung verfehlen.

2.2.2. Systeme mit nur einer Datenquelle

Lösungen mit nur einem Sensor verlassen sich stark auf dessen Genauigkeit – seine Fehlerraten und Ungenauigkeiten schlagen sich auch sehr direkt in den erzielbaren Positionswerten nieder (der Wanderer muss seinem Schrittzähler vertrauen). Eine eventuelle Nachverarbeitung der Sensordaten verfolgt primär das Ziel, ein gewisses Rauschen des Sensors zu eliminieren, Werte zu glätten und zu plausibilisieren, um etwa Sprünge oder plötzliche Richtungswechsel in der erhaltenen Positionsinformation zu vermindern (der Wanderer erkennt, dass sein Zähler übergelaufen ist und er die Differenz vom Start-Zählerstand bis zum Überlauf zu seiner Gesamtstrecke hinzu addieren muss).

Für solche Nachbehandlung der Daten kommen klassischer Weise Filtersysteme zum Einsatz, welche das Verhalten des zu ortenden Systems mathematisch zu modellieren versuchen und anhand dieses erwarteten Verhaltens eine Korrektur der Sensorinformation durchführen. Ihre Komplexität reicht von einfachen Schwellwert- und Bedingungsprüfungen bis hin zu komplexen Schätzalgorithmen wie dem Kalman-Filter, welcher versucht, *Vorhersagen* über das zu erwartende folgende Verhalten des Systems zu treffen und der als Erweiterter Kalmanfilter auch mit schwachen Nichtlinearitäten umgehen kann (siehe auch Kap. 2.2.4).

2.2.3. Systeme mit mehreren Datenquellen

Systeme, welche mehrere Informationsquellen verwenden, versuchen das unter Verwendung aller Informationen *beste erzielbare* Ergebnis zu liefern. Ein solches System kann im simpelsten Falle die Vertrauenswürdigkeit der Werte mehrerer Sensoren beurteilen und einen davon auswählen (der Wanderer vertraut seinem Schrittzähler *oder* dem Kompass), wird in der Regel jedoch eine *Fusion* der Werte anstreben und daraus eine *neue* (genauere) Positionsangabe errechnen (er nutzt wie oben beschrieben *beide* Geräte und die Karte).

Selbst bei Verwendung nur eines Sensors ist es möglich, weitere Informationsquellen in die Datengewinnung mit einzubeziehen: Map-Matching-Verfahren bilden Sensordaten unter Beachtung des jeweiligen Systemmodells auf existierende Karten ab, um Fehler korrigieren zu können (der Wanderer hat Schrittzähler und Karte). Auch dabei kommen oft zusätzliche Filter zum Einsatz, um Unschärfen in der erhaltenen Position zu vermindern (der Wanderer kennt die magnetische Missweisung in seiner Region und korrigiert seine Peilung).

Stehen die Daten *mehrerer Sensoren* zur Verfügung, ermöglicht dies ggf. eine weitere Verbesserung der Ortungsgenauigkeit, etwa, indem Systeme mit einer hohen kurzfristigen Genauigkeit aber mangelnden Langzeitstabilität (INS, Odometrie) mit Systemen mit langfristig konstanten Fehlern aber starken momentanen Unsicherheiten (etwa GNSS, logische

Ortung) kombiniert werden. Die Fusion solcher Daten verschiedener Sensoren ermöglicht die Verwendung noch weitergehender Filter- und Verrechnungsverfahren, die Verarbeitung von Ortungsinformationen einer wie auch mehrerer Quellen ist Gegenstand zahlreicher Veröffentlichungen und laufender Entwicklungen. Einen (nicht mehr ganz aktuellen) Überblick über Ortungs- und Navigationssysteme speziell im zivilen Fahrzeugbereich bietet [48], eine jüngere Abhandlung bei [58]. Einen Überblick über die Ortung speziell von Schienenfahrzeugen enthält [51].

2.2.4. Filterverfahren und Datenfusion

2.2.4.1. Vorbemerkungen

Messwerte – auch von Ortungssensoren – sind stets fehlerbehaftet. In jeder Messung können statistische wie systematische Abweichungen auftreten, Störgrößen mit der Messgröße korrelieren oder von ihr unabhängig sein. Einige Systeme weisen hohe kurzzeitige Unsicherheiten (stark schwankende Einzelwerte) auf, verfügen jedoch über eine gute Langzeitstabilität – andere wiederum können in kurzen Zeiträumen sehr präzise arbeiten, ihre Werten driften jedoch mit der Zeit oder werden immer unsicherer.

Um mit all diesen Einflüssen umgehen und sie in der Auswertung von Sensorsignalen berücksichtigen zu können, wurden zahlreiche Filtersysteme entwickelt. Zur weiteren Steigerung der Genauigkeit kommen außerdem Fusionsverfahren zur Anwendung, welche die Daten einer Quelle durch die aus einer weiteren zu ergänzen versuchen (siehe auch 2.2.3). Einige der für die Ortung wichtigsten Verfahren sollen im Rahmen dieses Abschnittes kurz vorgestellt werden.

2.2.4.2. Der Wiener-Filter

Der Wiener-Filter (auch: Wiener-Kolmogorow-Filter) ist ein linearer Optimalfilter für den Fall *stationärer* (d.h. im Mittelwert zeitlich unveränderlicher) Eingangssignale, die mit einem statistischen Rauschen behaftet sind. Er arbeitet mit der Gauß'schen *Methode der kleinsten Quadrate*. Falls im Rauschsignal Korrelationen vorliegen, müssen diese bekannt sein, um eine optimale Filterung erzielen zu können. Seinen Kern stellt die Wiener-Hopf'sche Gleichung dar – eine Integralgleichung, die nicht für alle Fälle gelöst werden kann und insbesondere in der numerischen Lösung durch Rechenanlagen erhebliche Probleme bereitet. Er spielt heute noch in der Bildverarbeitung eine gewisse Rolle, oft wird stattdessen jedoch ein Kalman-Filter eingesetzt [56].

2.2.4.3. Der Kalman-Filter

Der Kalman-Filter stellt eine direkte Weiterentwicklung des Wiener-Filters durch Rudolf E. Kálmán und Richard S. Bucy dar. Er kann auch mit *nichtstationären* (d.h. auch im Mittelwert zeitlich veränderlichen) Eingangssignalen umgehen und verwendet im Kern Differentialgleichungen (für kontinuierliche Signale) bzw. Differenzengleichungen (für diskrete), welche von Computern weitaus effizienter als die Integrale des Wiener-Filters zu lösen sind.

Der Kalman-Filter ist ein *Schätzer*, welcher auf Basis vergangener Zustände eines beobachteten Systems Vorhersagen über dessen zukünftiges (*erwartetes*) Verhalten trifft. Seine Schätzungen vergleicht der Filter permanent mit den real eingetretenen Fällen und passt die Parameter seiner Prädiktion entsprechend an. Dazu muss das Verhalten sowohl des beobachteten Systems als auch der zu erwartenden Fehlereinflüsse bekannt sein und in Form linearer Gleichungen modellhaft beschrieben werden. Ziel des Filters ist es, die Fehler (Störungen) aus dem Beobachtungssignal zu eliminieren und so auf den wahren Zustand des Systems (das Nutzsignal) schließen zu können.

Dabei handelt es sich beim Kalman-Filter eigentlich um gar keinen Filter im klassischen Sinne, sondern um einen rekursiv arbeitenden Algorithmus. Die Anpassung der Parameter der Gleichungen, welche Fehler und System beschreiben, erfolgt so, dass sich nach der *Methode der kleinsten Quadrate* die geringstmögliche Abweichung ergibt. Der Kalman-Filter schätzt für lineare Systeme *ideal*, d.h., dass kein anderer Filter bessere Vorhersagen treffen kann als dieser und er liefert stets den *wahrscheinlichsten* nachfolgenden Wert. Außerdem ist er asymptotisch stabil (er nähert sich mit fortschreitender Zeit immer näher dem wahren Wert an), seine Schätzungen sind wirksam, konsistent und erwartungstreu (die Varianz der Schätzungen ist minimal und strebt mit unendlich vielen Messungen gegen Null, ihr Mittel stimmt mit den tatsächlichen Werten überein) [47, S. 139]. Durch ein fortlaufendes Merken der bisher erreichten Zustände kann ein Kalman-Filter trotz bzw. dank seiner rekursiven Struktur sehr ressourcenschonend implementiert werden, weswegen er in seiner diskreten Form häufig auch in kleinen digitalen Systemen eingesetzt wird (es gibt aber auch eine kontinuierliche Form).

Bei all diesen vorteilhaften Eigenschaften unterliegt der originale Kalman-Filter jedoch einer Reihe wesentlicher Einschränkungen: Er kann ausschließlich mit linearen Systemen umgehen, nichtlineare Systeme können zu kontinuierlich falschen Prädiktionen oder schlimmstenfalls zur Divergenz des Filters führen. Außerdem muss es sich bei den zu eliminierenden Störgrößen (Rauschen der Ausgabe des Systems sowie Rauschen der aufgezeichneten Messwerte) im gegensatz zum Wiener-filter zwingend um *Weißes Rauschen* handeln. Zur Reduktion dieser Einschränkungen wurden Weiterentwicklungen oder Derivate des ursprünglichen Kalman-Filters entworfen. Die wichtigsten sollen nachfolgend kurz vorgestellt werden:

Beim *linearisierten Kalman-Filter* wird ein zu beobachtendes, an sich nichtlineares System um einen Arbeitspunkt herum linearisiert. Dazu wird die Systemfunktion in eine Taylorreihe entwickelt und diese nach dem linearen Glied abgebrochen. Zusätzlich wird ein vermuteter (initialer) Systemzustand außerhalb des eigentlichen Filters abgespeichert. Im Gegensatz zum herkömmlichen Kalman-Filter dient jedoch nicht das Systemmodell als Beschreibung, sondern ein analog entwickeltes Messmodell. Der Filter arbeitet nur auf diesem Messmodell, dessen Eingangswerte aus der Differenz von Messwert und vermutetem Systemzustand gebildet werden, sein Ergebnis wird anschließend mit Hilfe des (nicht mehr zum eigentlichen Filter gehörenden) Systemmodells korrigiert und auch der vermutete Systemzustand entsprechend angepasst. Es handelt sich damit um ein zweistufiges, indirektes Verfahren.

Der *erweiterte Kalman-Filter* (EKF) unterscheidet sich vom linearisierten dadurch, dass er direkt einen neuen vermuteten Systemzustand schätzt, anstatt mithilfe seines Ergebnisses eine Korrektur desselben vorzunehmen. Er bildet damit ein ebenfalls zweistufiges, aber direktes Verfahren. Beiden gemein ist, dass sie nicht mehr zwingend erwartungstreu sein müssen, unter starken Nichtlinearitäten der System- oder Messmodelle können sie sogar unzureichende Ergebnisse liefern oder divergieren. Dennoch kommen sie häufig zum Einsatz, da die beschriebenen Probleme für viele Anwendungsfälle nicht auftreten und der Aufwand zu ihrer Nutzung weit geringer als der für echte nichtlineare Filter ist (nach [47, S. 143 ff.]).

Darüber hinausgehend existieren noch weitere Anpassungen wie der Sigma-Point-Kalman-Filter, welcher mit stärkeren Nichtlinearitäten umgehen kann, oder der Kalman-Filter 2. Ordnung, welcher mittels mehrerer Iterationen durch das Messmodell die Leistung des EKF noch zu steigern versucht. Ebenso existieren Möglichkeiten vorhandene Korrelationen im System- und Messrauschen zu berücksichtigen und so die Beschränkung, dass der Kalman-Filter Weißes Rauschen als Störgrößen voraussetzt, zu umgehen – sowohl für den Fall bekannter Korrelationsfunktionen als auch unbekannter. Für diese und eine Vielzahl weiterer Vertiefungen sei auf die einschlägigen Kapitel in [47], [55] und [56] verwiesen.

Eine sehr kurze Einführung in die Verwendung des Kalman-Filters speziell bei GPS sowie der Kopplung von GPS mit INS bietet [54, S. 161 ff.], eine weit umfangreichere Darstellung auch verschiedener nichtlinearer Kalman-Filter findet sich bei [47, Kap. 6]. [56] stellt die Ableitung und Funktionsweise des ursprünglichen Kalman-Filters speziell für kontinuierliche Systeme detailliert dar, [57] beschreibt vorrangig den Entwurf und die praktische Anwendung des Filters am Beispiel klassischer Probleme der Luft- und Raumfahrt.

2.2.4.4. Markow-Modelle

Markow-Modelle (auch: Markof, Markoff, engl. Markov) dienen zur Gewinnung von Informationen über nicht direkt zugängliche Größen. Sie können auch als Filter bezeichnet werden, die Beobachtungen von Werten, welche mit den unzugänglichen Größen in Zusammenhang stehen, nutzen, um daraus Rückschlüsse auf die Werte selbst zu ziehen (Bsp.: Ein Bus kommt häufiger an den Tagen zu spät, an denen Schnee fällt, die Aufzeichnungen über Verspätungen enthalten aber keine Wetterdaten – Frage: An welchen Tagen fiel Schnee?).

Der Filter arbeitet dabei ausschließlich mit Wahrscheinlichkeiten: Er bedient sich eines *versteckten Markow-Modells* (Hidden Markov Model, HMM), welches das zu beobachtende System modelliert. Ein HMM besteht aus zwei Zufallsprozessen, von denen der erste (die eigentlich interessante unzugängliche Größe) durch eine Markow-Kette dargestellt werden kann, der zweite erzeugt mit einer gewissen Wahrscheinlichkeit die beobachtbaren Ereignisse. Die Größen der Markow-Kette an sich sind dabei nicht direkt zugänglich, sie sind verborgen (hidden) (die Kette simuliert die Wahrscheinlichkeit für Schneefall an einzelnen Tagen, der zweite Zufallsprozess beschreibt die Wahrscheinlichkeit, dass sich ein Bus bei Schneefall verspätet) [43].

Markow-Ketten sind Zustandsmodelle. Es bestehen Wahrscheinlichkeiten, dass ein Zustand in einen anderen übergeht, aber keine Ereignisse, die dies bewirken (wenn die Luftfeuchtigkeit bei Minusgraden sehr hoch ist, fällt mit 70%iger Wahrscheinlichkeit Schnee, zu 2% aber Eisregen, ansonsten bleibt es trocken). Die Wahrscheinlichkeit für einen Übergang kann durchaus auch 1 oder 0 sein, womit bestimmte Übergänge sicher eintreten oder nicht existieren. Handelt es sich nicht um diskrete Zustände (Schnee oder kein Schnee), sondern um kontinuierliche Zustandsräume, spricht man auch von Markow-Prozessen (wie dicht fällt der Schnee) [44].

Die Herausforderung bei Markow-Modellen liegt darin, zunächst einmal beobachtbare Parameter zu finden, welche mit der interessanten Größe in einem bekannten Zusammenhang stehen sowie eine Markow-Kette zu schaffen, welche das zu beobachtende System hinreichend genau beschreibt. Ihre Anwendung liegt daher auch weniger in der *Verarbeitung* von Werten eines Sensors, sondern vor allem darin, unter Verwendung gegebener und bekannter Modelle die *Verlässlichkeit* von Sensorwerten zu beurteilen (mit welcher Wahrscheinlichkeit ist die Position eines GPS-Empfängers genauer als x m, wenn y Satelliten sichtbar sind?).

2.2.4.5. Map-Matching

Unter dem Map-Matching oder dt. der Karteneinpassung versteht man eine Gruppe von Verfahren, welche das Ziel verfolgen, die Position eines Objektes (im allg. eines Fahrzeuges) straßen- oder schienengenau mit einer (digital) vorliegenden Karte in Einklang zu bringen. Dies ist etwa notwendig, um eine automatische Routenplanung in Navigationssystemen zu ermöglichen, Assistenzfunktionen in Fahrzeugen zu betreiben oder automatische Steuerungssysteme mit Informationen zu versorgen. Map-Matching kann aber auch verwendet werden, um die reine Ortungsgenauigkeit von Positionssensoren zu verbessern, da sich deren Fehlerverhalten durch den Vergleich mit Kartenmaterialien teilweise ausgleichen lässt. Dieser Abschnitt basiert fast vollständig auf [58], wo die hier erwähnten Verfahren sehr viel ausführlicher dargestellt und in ihrer Leistung und ihren Anwendungsgebieten miteinander verglichen werden.

Für landgebundene Fahrzeuge auf Straße und Schiene kommt häufig eines der folgenden fünf Verfahren zum Einsatz, die von [58] genauer analysiert wurden:

- Funktion über der Bogenlänge im Krümmungsbild
- Funktion über der Bogenlänge im Winkelbild
- Translation auf Koordinatenebene
- Ähnlichkeitstransformation auf Koordinatenebene
- Affintransformation auf Koordinatenebene

Diese Verfahren lassen sich in zwei Gruppen unterteilen: Die Funktionen über der Bogenlänge sind eindimensional. Die Funktionswerte beschreiben den Verlauf von Kurven und werden sowohl von den Ortungssensoren im Fahrzeug als auch aus dem Kartenmaterial ermittelt, es erfolgt eine Zuordnung der Fahrzeugposition zur Straße anhand bestmöglich passender Kurvenverläufe.

Die Funktionen auf der Koordinatenebene arbeiten zweidimensional. Sie zerlegen die gemessene (gefahrte) Trasse in diskrete Abschnitte und versuchen diese durch das jeweilige Transformationsverfahren bestmöglich auf gleichermaßen zerlegte Trassensegmente der digitalen Karte abzubilden.

Allen Techniken gemein ist, dass sie konkret ortsbezogene Informationen benötigen. Ortungsverfahren liefern ihre Werte im Allgemeinen jedoch in festen Zeitintervallen, so dass diese zunächst nicht zwingend auch für die Punkte vorliegen, für die sie auch benötigt werden (etwa den Beginn einer Kurve oder eines Trassensegmentes). Vor dem eigentlichen Map-Matching müssen daher ggf. noch Verfahren zur Datenaufbereitung zum Einsatz kommen [58, S. 9 ff.]. Einige exemplarische Möglichkeiten dafür werden im Folgenden

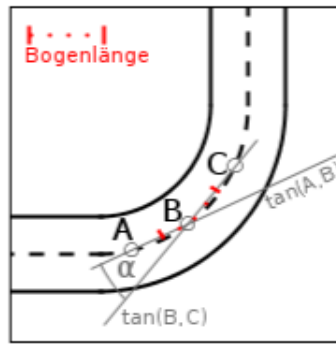


Abbildung 2.2.: Die Krümmung des Kurvenabschnittes ergibt sich aus dem Winkel zweier sich schneidender Geraden (Tangenten) sowie der zwischen ihnen liegenden Strecke.

zusammen mit den jeweils anwendbaren Methoden zum Vergleich der Mess- mit den Kartendaten dargestellt.

Zur Erzeugung des *Krümmungsbildes* wird eine Kurve durch mehrere kurze, von Stützpunkten begrenzte Geraden approximiert. Die Länge dieser Geraden wird so gewählt, dass diese nie mehr als eine vorgegebene Distanz von der Straßenmitte abweichen (z.B. 1/4 der Straßenbreite). In der Verlängerung entsprechen diese Geraden Tangenten (siehe auch Abb. 2.2, grau), welche an dieser Stelle an die Kurve angelegt sind. Eine Krümmung ist als Richtungsänderung je Winkeländerung definiert, so dass der Schnittwinkel α zweier Tangenten mit der zwischen ihren Berührungspunkten liegende Strecke (rot) jeweils einen Krümmungswert ergibt. Die so erhaltenen Werte werden anschließend durch eine differenzierbare Funktion approximiert. Nach der Bogenlänge differenziert ergibt sich damit die Krümmung innerhalb der Kurve als Funktion der Bogenlänge (d.h. der in der Kurve zurückgelegten Wegstrecke; Abb. 2.3 stellt diese beispielhaft für noch nicht approximierten Werte dar). Die Positionsinformationen von Sensoren können auf ähnliche Weise dargestellt werden. Bei GPS-Werten werden z.B. ebenfalls die Schnittwinkel zweier benachbarter Geraden bestimmt und nach dem gleichen Verfahren in eine Funktion abhängig von der Wegstrecke überführt.

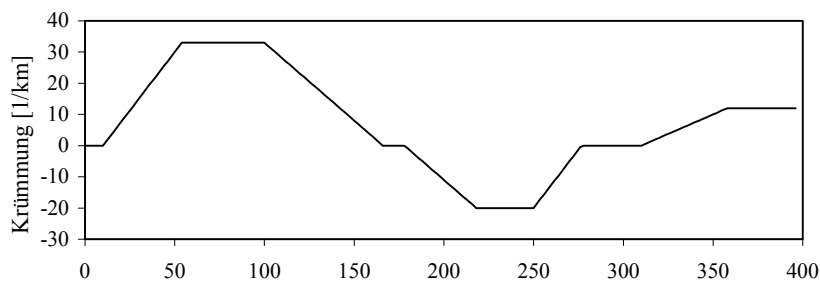


Abbildung 2.3.: Die nicht approximierten Krümmungswerte einer Kurve als Funktion der Bogenlänge. Bildquelle: [58, S. 17]

Für das *Winkelbild* kommen ähnliche Verfahren wie beim Krümmungsbild zur Anwendung. Ausgehend von den Stützpunkten einer Kurve wird jedoch direkt der Richtungs-

winkel ihrer Teilgeraden erfasst (Abb. 2.4, graue Linie). Durch eine entsprechende Approximation des Verlaufes können auch diese nach der Wegstrecke differenziert und damit als Wert (Winkel) in Abhängigkeit von der Bogenlänge dargestellt werden (Abb. 2.4, rote und schwarze Linie).

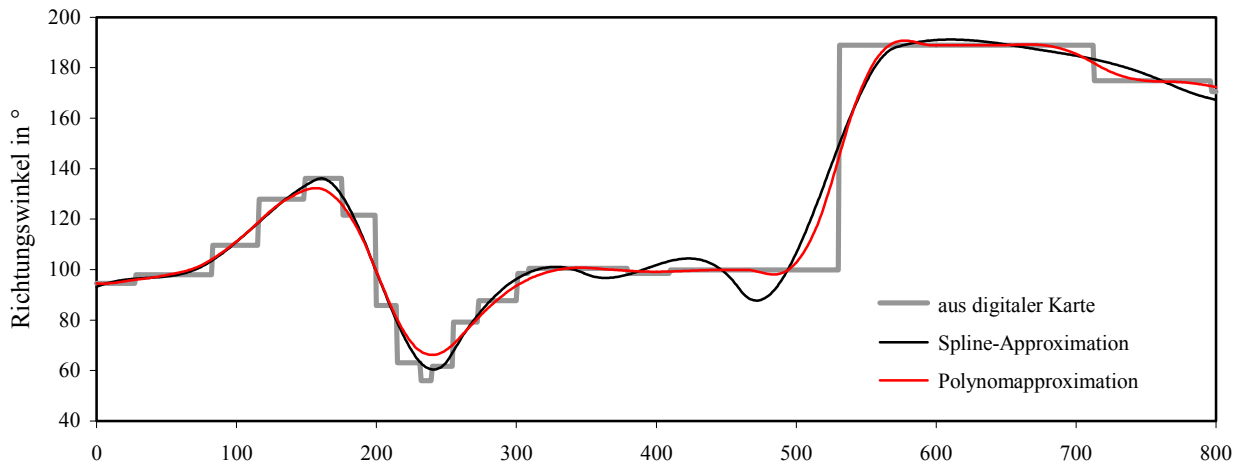


Abbildung 2.4.: Richtungswinkel einer Kurve als Funktion der Bogenlänge, vor und nach der Glättung. Bildquelle: [58, S. 26]

Die so entstandenen Funktionen liefern für die digitale Karte sowie die Messwerte im theoretischen Idealfall für identische Bogenlängen x auch identische Werte y . Praktisch müssen dabei immer Ungenauigkeiten ausgeglichen werden, wofür eine ganze Reihe verschiedener Verfahren zur Verfügung stehen. Eine Möglichkeit besteht darin, die Ähnlichkeit der Funktion gemessener Werte mit der Funktion der Kartendaten über deren Kreuzkorrelation zu bestimmen. Dazu werden die Funktionen zunächst normiert (etwa auf einen Wertebereich von ± 1 sowie Mittelwertfreiheit), um Skalierungsprobleme zu umgehen (der Maßstab der verwendeten Daten zueinander ist jetzt bedeutungslos), dann wird ihre Kreuzkorrelation gebildet. Diese wird nie einen Einheitsimpuls ergeben, da die Signalformen in der Praxis nie völlig identisch sind. Ihr Maximum ist jedoch ein gutes Maß für die Ähnlichkeit beider Kurven. Werden alle in einem Suchgebiet liegenden Kurven der Karte auf diese Art mit der gemessenen Kurve verglichen, lässt sich feststellen, welche Kurve das Fahrzeug tatsächlich gefahren ist (die, deren Kreuzkorrelation das höchste Maximum aufweist). Das Ergebnis ist umso aussagekräftiger (wahrscheinlicher), je größer das größte Maximum von den übrigen entfernt ist. Hat eine Strecke nur sehr wenige oder nur sehr leichte Kurven oder haben mehrere in Frage kommende Kurven einen sehr ähnlichen Verlauf bei ähnlicher Orientierung, so kann dieses Verfahren allein gegebenenfalls keine hinreichend sicheren Aussagen treffen. Aufgrund seiner Abhängigkeit von Weglängen und Winkeln wird es vorwiegend mit Sensorkombinationen eingesetzt, die direkt entsprechende Werte liefern, etwa (Differential-)Odometern, Magnetfeldsensoren (elektr. Kompass) und Inertialsensoren.

Die Gruppe der Funktionen auf *Koordinatenebene* arbeitet anders. In ihnen werden die Trassen der digitalen Karte in Form diskreter Punkte mit konstantem Abstand dargestellt (Größenordnung: wenige Meter). Der Vergleich zwischen den gespeicherten und den gemessenen Trassen erfolgt über diese diskreten Punkte. Die Messwerte verwendeter Ortungssensoren müssen daher vor ihrer Verwendung zum matching durch Interpolation auf Streckenabschnitte identischer Länge angepasst werden.

Die aus den Messungen interpolierten Wegpunkte werden über jeweils eine der drei angegebenen Transformationsfunktionen auf die Kartendaten abgebildet und deren Übereinstimmung geschätzt. Anschließend werden sie um einen Kartenpunkt verschoben und die Übereinstimmung erneut geschätzt. Nach Iteration über alle in Frage kommenden Wegpunkte ergibt sich die Position des Fahrzeuges als die Abbildung mit der höchsten aufgetretenen Wahrscheinlichkeit (die Position mit der am höchsten eingeschätzten Übereinstimmung). Der Unterschied der drei Verfahren liegt darin, wie diese die Abbildung der Messdaten auf die vorliegenden Kartendaten zulassen. Die *Translation* ermöglicht nur die Verschiebung der gemessenen Werte, um diese mit der Karte in Übereinstimmung zu bringen. Die *Ähnlichkeitstransformation* erlaubt weiterhin Drehungen und maßstäbliche Vergrößerungen oder Verkleinerungen des gemessenen Abschnittes, die Affinitätstransformation außerdem noch Scherungen, Spiegelungen und Projektionen. Entsprechend muss die Abschätzung der Übereinstimmung für jedes Verfahren angepasst werden, sie alle basieren jedoch darauf, die Varianz der jeweiligen Iteration zu berechnen und am Ende den Iterationsschritt mit der niedrigsten aufgetretenen Varianz zu bestimmen (nach: [58, S. 10 sowie 69 ff.]).

2.2.4.6. SLAM

Simultaneous Localization and Mapping (SLAM) bezeichnet ein Problemfeld insbesondere der modernen Robotik: Wenn über eine Umgebung zunächst keinerlei Informationen vorliegen, muss sich ein Roboter darin sowohl ein eigenes Bezugssystem schaffen und in diesem orten (Localisation) als auch Informationen über die Umgebung sammeln, um seine eigene Position in dieser fortwährend wiederfinden und darin navigieren zu können (Mapping).

In den Komplex der Ortung fallen dabei viele der bisher bereits diskutierten Techniken. Sie alle verfolgen letztendlich das Ziel, auf Grundlage der vorhandenen (Sensor-)Informationen Aussagen darüber zu treffen, wo sich das zu ortende Objekt im jeweiligen Bezugssystem befindet. Dabei existieren zwei wesentliche Gruppen von Ansätzen: Solche, die definierte Koordinaten angeben und solche, welche mit Wahrscheinlichkeiten arbeiten.

Die erste Gruppe fusioniert Daten mit dem Ziel, den in Frage kommenden Bereich des Aufenthaltsortes so weit einzuschränken, dass ein *bestmöglich bestimmter Punkt* als „sicherer“ Ort angegeben werden kann. Dabei ist es nicht zwingend erforderlich (oder möglich),

dass dieser bestmöglich bestimmte Ort auch dem tatsächlichen entspricht - er dient jedoch als der temporäre Referenzpunkt, von dem aus alle weitergehenden Entscheidungen getroffen werden (Routenplanung, Map-Matching, Verknüpfung bereits vorhandener mit neu gewonnenen Karteninformationen, Filterverfahren zur Wegfortschreibung, ...).

Die zweite Gruppe verzichtet auf derartige Festlegungen. Sie arbeitet mit *Aufenthaltswahrscheinlichkeiten* für jede in Frage kommende Position. Der Vorteil dieses Ansatzes liegt darin, dass es mehrere Orte mit sehr ähnlich hoher Wahrscheinlichkeit geben kann – der Roboter kann schlicht keine verlässliche Aussage darüber treffen, wo er sich denn tatsächlich befindet. Damit ist er nicht gezwungen, sich für einen dieser Punkte zu entscheiden und die Informationen über andere Orte zu verwerfen. Nachteilig ist dabei jedoch, dass auch die o.g. Entscheidungen nicht mehr ohne weiteres getroffen werden können – wenn es keinen „sicher“ bekannten Startpunkt gibt, kann von diesem aus auch keine Routenplanung erfolgen. Eine Nutzung sämtlicher Verfahren auf allen in Frage kommenden Punkten ist nicht immer praktikabel, da dies einerseits in der Praxis die verfügbare Hardware überfordern könnte, andererseits aber aus den dann entstehenden zahlreichen Entscheidungsmöglichkeiten immer noch eine ausgewählt werden müsste, um eine konkrete Handlung durchzuführen (welche von mehreren gleich wahrscheinlichen Routen zu einem Ziel soll gefahren werden?).

Den zweiten Komplex bildet das Mapping: Für viele Anwendungen wird nicht nur eine Lokalisation benötigt, mit der sich gefahrene Routen wie auch vergangene und aktuelle Punkte numerisch erfassen lassen, sondern das System benötigt zusätzlich Informationen über seine Umwelt – eben eine Karte derselben, mittels der markante Stellen, zurückgelegte Strecken und ortsbezogenes Wissen abgebildet werden können. Ohne eine Karte ist etwa nie eine sinnvolle Routenplanung möglich, da keine Informationen über Hindernisse auf der Strecke vorliegen, keine Korrektur von Routen durch Karteneinpasung (Map-Matching) und auch keine Justierung driftbehafteter Sensorik (IMU, Odometrie), so dass sich Sensorfehler mit der Zeit immer weiter kumulieren. Im Rahmen dieser Arbeit tritt das Problem des Mappings nicht auf, da gegebenenfalls genutztes Kartenmaterial stets bereits vorhanden ist und die Sensorik nicht dafür ausgelegt wurde, die für eine Kartenerstellung notwendigen Informationen zu erfassen. Aufgrund der Bedeutung von Kartenmaterialien und ihrer engen Verknüpfung mit einigen der bereits erwähnten Verfahren zur Genauigkeitverbesserung wie auch zur Auswertung erfasster Daten sollen die wichtigsten Mapping-Verfahren im Folgenden dennoch kurz vorgestellt werden:

Occupancy Grid Mapping beschreibt das Anlegen einer Karte auf Basis eines Rasters. Ausgehend von Daten über die Orientierung des Systems und Messwerten von hinderniserkennenden Sensoren (etwa Laserscanner, Ultraschall, Infrarotentfernungsmesser, Berührungstaster) wird jedem Punkt dieses Rasters eine binäre Information zugeordnet: Belegt oder nicht belegt. Damit entstehen Karten, welche die Umgebung als eine Ansammlung von freien Flächen und Hindernissen abbilden. Direkt von einem System erfasste Messwerte

sind dafür oft nicht zu gebrauchen, da die Position und Orientierung (Pose) des messenden Systems nicht hinreichend präzise erfasst werden können und auch die Entfernungssensoren einem gewissen Fehler unterliegen. Das Occupany Grid Mapping kommt daher oft nicht direkt während eines Einsatzes, sondern erst nach anderen kartenerzeugenden Verfahren zum Einsatz, um das bereits gewonnene Material in einer für Roboter und andere praktische Anwendungen nutzbaren Form darzustellen [60]. Auf den dabei letztendlich entstehenden, rasterorientierten Kartendaten können gegebenenfalls weitere Algorithmen laufen, um etwa Konturen von Objekten zu schließen, als unzugänglich erkannte Bereiche auszublenden oder einzelne Fehlerstellen zu bereinigen.

Muss das Erstellen einer Karte noch während des Laufes erfolgen und hat das System keinerlei externe Referenzen zur Verfügung, um etwa seine Position und Ausrichtung präzise bestimmen zu können, wird das Mapping erheblich schwieriger (*Simultaneous Localization and Mapping*). Jede zukünftige Pose muss jetzt aus der vorherigen abgeleitet werden, um Messwerte zu Hindernissen und Freiflächen noch korrekt zuordnen zu können. Dafür gibt es nach [60] zwei wesentliche Ansätze: *online* SLAM und *full* SLAM. In ersterem wird zur Gewinnung einer neuen Positionsinformation immer nur die letzte herangezogen, so dass sich eine Fortschreibung sämtlicher Parameter ergibt. Dadurch drängen sich Verfahren zur Verarbeitung und Korrektur dieser Verfahren auf, welche mit diskreten, miteinander stark korrelierten Informationen umgehen können, ohne dabei Zugriff auf eine entsprechende Historie der einzelnen Werte zu haben (etwa Wiener- und Kalman-Filter). Full SLAM beschreibt im Gegensatz dazu die Idee, die gesamte Geschichte der bisherigen Positionierung in die Gewinnung der nächsten Information mit einzubeziehen: Die neue Pose wird aus allen bisherigen und dem Verlauf ihrer Entstehung ermittelt. Dieser Ansatz bietet prinzipiell bessere Möglichkeiten zur Gewinnung präziserer Informationen, leidet aber an seiner schnell zunehmenden Komplexität und der Unmöglichkeit, beliebig große Datenmengen entsprechenden Filter- und Berechnungsverfahren zu unterziehen. Informationen zu Grundlagen wie auch konkreten Algorithmen für beide Fälle finden sich in [60, Part III].

2.3. Ortungssysteme für Fahrzeuge

Die technologisch unterstützte Ortung von Fahrzeugen bzw. allgemein die sichere Positionsbestimmung an Land ist ein relativ junges Gebiet. Erste Ansätze dafür sind bereits seit der Antike vorhanden, eine hohe Verbreitung erreichten die heute bekannten Systeme jedoch erst innerhalb der letzten 20 Jahre. Treibende Kraft hinter vielen modernen Anwendungen war das militärische Bedürfnis nach sicherer Positionsbestimmung und Navigation, welches seit den Erfahrungen des Zweiten Weltkrieges und der anschließenden Episoden des Kalten Krieges unter sämtlichen Industrienationen erwuchs. Die Resultate

dieser Bestrebungen waren für zivile Anwendungen i.d.R. jedoch nicht verfügbar, was die Entwicklung in diesem Bereich bis zum Zusammenbruch des Ostblockes und zum damit einhergehenden Ende des Kalten Krieges lähmte.

Erste Ansätze wie das mit Radiowellen arbeitende Hyperbelnavigationssystem Decca konnten nur regional und mit einem hohen infrastrukturellen Aufwand genutzt werden, an einen Masseneinsatz wie in der heutigen Zeit war noch nicht zu denken. Bei diesem und ähnlichen Systemen senden mehrere räumlich getrennte Sender Radiowellen bekannter Frequenz aus, die miteinander interferieren. Ihre Phasendifferenz ist ein Maß für den eigenen Standort, welcher auf einer Hyperbelstandlinie zwischen den Sendern liegen muss. Bei Empfang von mindestens drei Wellen lässt sich so eine Positionsangabe berechnen (Schnittpunkt zweier Hyperbeln) ([53], [1]). Auch nachfolgende Systeme arbeiteten zu meist mit terrestrischen Funkstationen (Omega, Loran) oder optischen Signalbaken, was für lange Zeit zum einen der Reichweite und Genauigkeit solcher Systeme gewisse Grenzen setzte, zum anderen aber vor allem aufgrund der damit verbundenen hohen Anschaffungs- und Betriebskosten eine breitere Nutzung verhinderte.

In öffentlichen Verkehrsmitteln kommt daher seit den 1970er Jahren oft eine einfachere Methode zum Einsatz: Der Fahrer signalisiert per Knopf- oder Pedaldruck das Erreichen bzw. Passieren einer Haltestelle. Die Fahrzeugelektrik nutzt dieses Signal etwa zum Umschalten vordefinierter Anzeigen und meldet diese „letzte bekannte Position“ auch regelmäßig per Funk an die Zentrale des Verkehrsunternehmens. Zusammen mit einer aktuellen Karte der geplanten Fahrzeugroute sowie der darauf liegenden Haltestellen lässt sich somit die Position eines Fahrzeuges zumindest auf einen Streckenabschnitt eingrenzen – ein System, welches bis heute noch oft im *öffentlichen Personennahverkehr* (ÖPNV) bei Bussen und Straßenbahnen zum Einsatz kommt.

Ergänzt wird diese Idee der „logischen Ortung“ durch technische Hilfen wie Odometer, welche die zurückgelegte Wegstrecke ab einem definierten Nullpunkt (etwa dem Linienanfang oder einer Kehrschleife) erfassen und damit unter Berücksichtigung des Linienplanes Rückschlüsse auf die aktuelle Position ermöglichen. Türöffnungssignale wurden und werden zur Erkennung von Haltepunkten und somit etwa zur Korrektur von odometrisch arbeitenden Systemen genutzt. Vor allem im Schienenverkehr kamen auch ortsfeste Systeme wie Infrarot- oder Magnetbaken oder induktiv arbeitende Koppelspulen zum Einsatz, welche aber einen hohen zusätzlichen infrastrukturellen Aufwand erforderten und nur sehr bedingt für den Straßenverkehr geeignet waren [50, Kapitel 4].

Im regulären Eisenbahnwesen gibt es außerdem das aus der Streckensicherung stammende Konzept der Blocksicherung, welches bis heute häufig anzutreffen ist – eine Strecke wird in mehrere Blöcke unterteilt, deren Frei- oder Belegtmeldung i.d.R. technisch erfasst und an eine höhere Leitebene – etwa ein Stellwerk – geleitet wird. Dort lässt sich verfolgen, welchen Block ein Zug gerade passiert, womit sich seine Position ähnlich dem vorherigen

Verfahren zumindest auf einen Bereich zwischen zwei Punkten eingrenzen lässt. Diese Methode spielt bis heute eine große Rolle, zumal sie als „Nebenprodukt“ der oft notwendigen Streckensicherung ohnehin zur Verfügung steht [59, S. 7 ff.].

All diesen Systemen gemeinsam ist jedoch der Nachteil, dass sie kaum für *individuelle* Anliegen zu gebrauchen sind – sie ermöglichen keine beliebige Verwendung an zuvor nicht explizit dafür ausgestatteten Orten. Diese Situation änderte sich erst 1993 mit dem Erreichen der initialen Betriebsbereitschaft des amerikanischen *Global Positioning Systems* (GPS). Dessen Vorläufer *Transit* war zwar bereits seit 1964 in Betrieb, jedoch erst ab 1967 für zivile Anwendungen verfügbar und erlaubte je nach den momentanen Positionen der Satelliten nur eine mehrere Minuten in Anspruch nehmende Peilung im Zeitabstand von ein bis zwei Stunden (schlechter als 100 m) [53, Kapitel 11.3], [54]. Seit der Abschaltung der künstlichen Verschlechterung (*Selective Availability, SA*) des zivilen GPS-Signales im Frühjahr 2000 erlebte das Segment insbesondere mobiler und kompakter Empfangsgeräte einen enormen Aufschwung [45, S. 3], das System erreicht eine globale Verbreitung wie kein anderes Projekt zuvor. Damit stehen heute auch für Low-Cost- und Massenanwendungen hochintegrierte Empfänger mit einer gewissen Genauigkeit zur Verfügung [2, 2. und 4. Absatz].

Neben dem GPS gelangen derzeit eine Reihe weiterer *Globaler Navigationssatellitensysteme* (GNSS) zur Einsatzreife, darunter etwa das russische ГЛОНАСС (*GLONASS*, Global Orbiting Navigation Satellite System), welches bereits 1996 einmal einsatzfähig war (nach [54, 2.4] bereits 1992) und seit einigen Jahren wieder aufgebaut wird [47, S. 83], das chinesische Compass oder das europäische Galileo [45, S. 4 ff.].

Abseits der satellitengestützten Ortung haben sich in den letzten Jahren zwei weitere wichtige Felder aufgetan: Die Positionsbestimmung über Mobilfunknetzwerke sowie die über momentan sichtbare WLAN-Funkzellen. In beiden Verfahren ermittelt das Empfangsgerät alle in seiner Reichweite befindlichen Funkzellen und gleicht diese Liste mit bestehenden Datenbanken ab, um anhand der Funkzellengrößen und ggf. noch der empfangbaren Signalstärken seine ungefähre Position errechnen zu können. Diese Verfahren sind stark von der Verfügbarkeit und Größe ausreichend vieler Funkzellen und der dazugehörigen aktuellen Datenbanken abhängig, wodurch ihre Genauigkeit von wenigen Metern bis hin zu mehreren Kilometern schwanken kann. Der technologisch tendenziell höheren Genauigkeit der WLAN-basierten Ortung steht der private und unkontrollierte Betrieb der meisten Access Points gegenüber, welcher Datenbanken schnell veralten lässt und auch signifikante Fehlerraten produzieren kann. Zudem besteht oft nur in dicht bebautem städtischem Gebiet eine ausreichende Funkabdeckung. Mobilfunkzellen hingegen werden nur von einer überschaubaren Anzahl großer Provider betrieben, wodurch beständig aktuelle Daten über diese zur Verfügung stehen. Dieses Verfahren bietet auch in ländlichen Gegenden oft zumindest noch eine Basisabdeckung, ist aufgrund der größeren Zellen aber oft auch ungenauer.

2.4. Kommunikation und Datenfunk im ÖPNV

2.4.1. Analogfunksysteme

Die Geschichte des Funkverkehrs im deutschen ÖPNV beginnt in den 1960er Jahren: Damals etablierten sich erste analoge Informations- und Sprechfunkdienste, welche jeweils von einer lokalen oder regionalen Leitstelle aus betrieben wurden [3]. *Informationsfunk* meint in diesem Zusammenhang explizit noch keine Datenübertragungsdienste im heutigen Sinne, sondern einfache Positionsrückmeldungen wie das Passieren vorgegebener Punkte nach der in Kapitel 2.3 beschriebenen Methode. Zur Blüte gelangten diese Systeme Mitte der 1970er Jahre, als die Analogfunktechnik einen immer ausgereifteren Stand erreichte.

Die auf diese Technologie zurückgehenden Funkssysteme insbesondere im 70-cm-Band wurden später um einem schmalbandigen Datenfunk ergänzt und werden im ÖPNV bis heute noch sehr oft eingesetzt. Viele regional agierende Verkehrsunternehmen verfügen nach wie vor über einige wenige Sprech- und Datenfunkfrequenzen, welche im Zeitmultiplexverfahren zwischen allen mobilen Einheiten und der Zentrale geteilt werden müssen. Weiterentwicklungen dieser analogen Funkssysteme fanden nur noch begrenzt statt. Erwähnung verdient hierbei der Standard *MPT 1327/1343* des britischen Ministeriums für Post und Telekommunikation, welcher ein Bündelfunksystem (siehe 2.4.2.4) auf Basis analoger Übertragung realisiert und auch in Deutschland als BOS-Funk und Alarmierungsnetz in Gebrauch war oder ist [52]. Obwohl einstmals explizit für den Gebrauch mit *rechnergestützten Betriebsleitsystemen* (RBL) entwickelt, sind diese analogen Systeme den heutigen Anforderungen an Reichweite, Sprachqualität und Datenübertragungsmöglichkeiten nicht mehr gewachsen.

2.4.2. Digitalfunksysteme

2.4.2.1. Überblick

Seit rund 20 Jahren verstärken sich die Bestrebungen, den mittlerweile als überholt angesehenen Analogfunk durch digitale Systeme zu ersetzen. Die dafür eingesetzten Lösungen unterscheiden sich im ÖPNV nicht von anderen Betriebsfunksystemen, wohl aber von herkömmlichen Mobilfunknetzen. Zur Nutzung der *GSM*-Technologie gab es durchaus Anläufe – die öffentlichen D- und E-Netze konnten den Anforderungen der Verkehrsunternehmen jedoch nicht gerecht werden und dedizierte Netze wurden nur für den (Eisen-)Bahnbetrieb in Form von *GSM Railway* bis zur Anwendungsreife entwickelt. Insbesondere die Forderungen nach einer *sicheren Verfügbarkeit*, *garantierten Mindestbandbreiten* sowie einer begrenzten *Unabhängigkeit von externer Infrastruktur* (etwa durch

Peer-to-Peer-Kommunikation zwischen den Endgeräten) führen bislang dazu, dass die Verkehrsunternehmen auch bei neu aufzubauenden Systemen auf eigene Funknetze setzen, anstatt auf bestehende, mit anderen Teilnehmern geteilte Lösungen zurückzugreifen. Einen kurzen Abriss funkbasierter und weiterer Technologien zur Kommunikation und Datenübertragung für Verkehrssysteme bietet [50, Kapitel 5.3].

2.4.2.2. DMR

Im digitalen Betriebsfunk kleinerer Bedarfsträger (Gebietsabdeckung mit nur einer Funkzelle) sind derzeit vor allem die beiden ähnlich gelagerten Systeme *Digital Mobile Radio* (DMR) und *digital Private Mobile Radio* (dPMR) anzutreffen.

DMR ist ein Zeitmultiplexverfahren, welches zwei Slots bei 12,5 kHz breiten Kanälen bietet und damit effektiv einer Vollduplexübertragung für Sprache und Daten mit 6,25 kHz / Übertragungsrichtung gleichkommt. Das vom *European Telecommunications Standards Institute* (ETSI) standardisierte Verfahren erreicht bis zu 9,8 kBit/s brutto und richtet sich primär an kleine und mittelgroße kommerzielle Anwender [6]. Es unterstützt u.a. einen schnellen Gesprächsaufbau, Gruppenrufe, Rundrufe (Broadcast), Priorisierung und Notrufe sowie eine Direktkommunikation zwischen Endgeräten (Peer-to-Peer) [5, 6. Block]. In Deutschland vergibt die Bundesnetzagentur (BNetzA) Frequenzgenehmigungen für das 2-m- sowie das 70-cm-Band [4]. DMR-Funkgeräte sind oft abwärtskompatibel zu bestehenden analogen Lösungen im gleichen Frequenzbereich, was große Vorteile bei der Migration bestehender Installationen mit sich bringt – es können Alt- und Neugeräte zusammenarbeiten und vorhandene Lizenzen der abzulösenden Analogfunkspektren weiter genutzt werden. Ein Aufbau mit mehreren Funkzellen hingegen ist zwar denkbar, aber nicht vorgesehen [7, 5. bis 7. Absatz].

2.4.2.3. dPMR

dPMR ist ein ebenfalls vom ETSI standardisiertes Frequenzmultiplexverfahren und technologisch gesehen ein direkter Konkurrent zu DMR. Es verwendet 6,25 kHz breite Kanäle, von denen jeweils zwei direkt benachbarte zu einem 12,5 kHz-Kanal zusammen geschaltet werden können und bietet je (6,25 kHz-)Kanal 4,8 kBit/s Brutto-Datenübertragungsrate [8]. Die Frequenzbänder sollen ebenfalls wie beim DMR identisch zu den bisher analog genutzten sein können, lediglich eine Abwärtskompatibilität zum echten Analogfunk ist nicht vorgesehen. Die Zielgruppe beider Systeme ist ebenfalls beinahe gleich – dPMR zielt auf fortgeschrittene private, vor allem aber kommerzielle Anwender vom kleinen Betriebsfunk bis hin zu „Mission-Critical applications“, wie es das ETSI in [9, Seite 2] selbst beschreibt. dPMR ist in Spezifikation und Verbreitung noch nicht ganz so weit fortgeschritten wie DMR – viele Dokumente tragen noch den Vermerk „draft“ oder „proposed standard“ – ist aber ebenfalls bereits auf dem freien Markt erhältlich.

Aufgrund der gemeinsamen Herkunft und der überschaubaren Zahl von Lieferanten für Komponenten beider Systeme herrscht hier keine klassische marktwirtschaftliche Wettbewerbssituation zwischen den Herstellern, sondern eher ein Entscheid durch die Kunden, dessen Ausgang bislang noch nicht abzusehen ist. Die aus dem 2007 begonnenen *dPMR Memorandum of Understanding* hervorgegangene *dPMR Association* hat eine erläuternde Webseite eingerichtet [10], in deren F.A.Q.-Sektion die Lage auf den Punkt gebracht wird:

„Q4: Which is better, 6.25 kHz FDMA or 12.5 kHz TDMA?

A4: This is the million dollar question. The answer is that both technologies have been accepted in the market based on the features and advantages provided respectfully. This basically says there is room for both and it is still an even race.“ [11, Frage 4]

Vorerst müssen also beide Systeme als gleichwertige Alternativen betrachtet werden, wenngleich dies im Rahmen dieser Arbeit keine unmittelbaren Auswirkungen hat.

2.4.2.4. TETRA

Mit dem ebenfalls vom ETSI standardisierten *Terrestrial Trunked Radio* (TETRA) befindet sich noch ein weiteres digitales Funksystem am Markt. Dieses ist deutlich leistungsfähiger als die vorgenannten Systeme ausgelegt und bietet eine Reihe darüber hinausgehender Funktionen. Insbesondere ist es explizit als Bündelfunksystem konzipiert – d.h., dass sich mehrere (durchaus verschiedene, voneinander unabhängige) Nutzergruppen eine gemeinsame Funkinfrastruktur und gemeinsame Frequenzen teilen können und diese erst auf der Protokollebene in die einzelnen logischen Kanäle aufgeteilt werden. Diese Art der Nutzung ist zwar prinzipiell auch mit DMA oder dPMA denkbar, führt diese Netze aufgrund der schmalen Kanäle und geringeren Übertragungsraten jedoch schnell an ihre Grenzen. Die Trennung der TETRA-Nutzergruppen kann mittels einfacher Telegrammkennungen geschehen, oft kommen jedoch vollwertige digitale Verschlüsselungsverfahren und teilweise auch Benutzerauthentifizierungssysteme zum Einsatz. Ebenso möglich ist der Aufbau eines weitreichenden Netzwerkes mit Roaming und Handover zwischen den einzelnen Funkzellen, zentraler Administration und Verwaltung (Zellgrößen bis 25 km, [52]). Hinzu kommen Spezialanwendungen wie die Nutzung mobiler Endgeräte als Relaisstationen zur temporären Reichweitensteigerung oder der Aufbau beliebig (teil-)vermaschter Netzwerke in Situationen, in denen keine Infrastruktur vorhanden, diese überlastet oder zerstört ist. TETRA zielt damit explizit auf den Ersatz der teils immer noch analogen Funknetze von *Behörden und Organisationen mit Sicherheitsaufgaben* (BOS) sowie auf Nutzergruppen mit erhöhten Anforderungen an die Reichweite, Belastbarkeit und Sicherheit ihrer Funkinfrastrukturen [12], [52].

TETRA beherrscht verschiedene Modulationsverfahren auf je nach Anwendung 25 bis 150 kHz breiten Kanälen, wobei jeder dieser Kanäle in eine Up- und eine Downlinkfrequenz

unterteilt sowie im Zeitmultiplexverfahren belegt wird. Dies ermöglicht Bruttodatenraten bis nahezu 700 kBit/s (bis zu 300 kBit/s im Realeinsatz mit TETRA Enhanced Data Service). Am weitaus häufigsten werden dabei 25 kHz breite Kanäle mit einer $\pi/4$ -DQPSK-Modulation eingesetzt, was bei mittlerer Güte der Kanalcodierung bis zu 19,2 kBit/s netto ergibt (36 kBit/s brutto) [52, S. 16 f.]. Für eine einzelne Sprechverbindung werden bei Vollduplex 7,2 kBit/s genutzt (4,75 kBit/s je Richtung) [13, Abschnitt 2.3], [14, vorletzter Absatz]. Dieser qualitativ weitaus höherer Ansatz als bei den übrigen Systemen ermöglicht entsprechend ressourcenintensive Anwendungen, allerdings um den Preis eines deutlich gesteigerten Frequenzverbrauches. TETRA-Systeme belegen daher auch nicht die klassischen Analogfunkfrequenzen, sondern nutzen andere, explizit dafür vorgesehene Frequenzräume bei 380, 470 und 870 bis 933 MHz (TETRA-BOS in Deutschland etwa 380–395 MHz). [13, insb. Abschnitte 2.2 und 2.7]. Eine Verwendung auf 1.8 GHz ist geplant [52, S. 16 f.].

Im privatwirtschaftlichen Umfeld nutzen derzeit vorrangig größere Gesellschaften TETRA-Funknetze, darunter eine Reihe von Flughafenbetreibern [16], [17], [18] und Versorgungsunternehmen [15]. Auch mehrere Verkehrsdienstleister haben diese Technologie für sich adaptiert und setzen sie zur Abdeckung großer Stadtgebiete oder Ballungsräume ein, die mit einfacheren Verfahren nicht mehr zu handhaben sind [19, Abschnitt „Akteure“].

2.4.2.5. TETRAPOL

Bei *TETRAPOL* handelt es sich um ein proprietäres, ursprünglich von der französischen Matra-Gesellschaft in Kooperation mit EADS für die französischen Sicherheitskräfte entwickeltes Bündelfunksystem [20, Abschnitt „Entwicklung“]. Seit der Übernahme weiterer Teile von Matra durch den einstigen Partner war EADS-Telecom längere Zeit der einzige Anbieter von TETRAPOL-Komponenten. Inzwischen strebt die EADS jedoch mittels des TETRAPOL-Forums eine internationale Standardisierung an und hat Teile der technischen Spezifikationen offen gelegt bzw. an interessierte Unternehmen herausgegeben, welche somit ebenfalls als Zulieferer in Erscheinung treten können [23]. Die EADS tritt im Zusammenhang mit TETRAPOL heute zumeist über ihre stark militärisch orientierte Tochtergesellschaft *Cassidian* bzw. deren Bereich „Communication Network Security“ in Erscheinung, sowohl im TETRAPOL Forum [22] als auch in der (militärischen wie zivilen) Vermarktung der TETRAPOL-Systeme [21]. Neben TETRAPOL-Komponenten [24] ist das Unternehmen auch im konkurrierenden TETRA-Segment tätig [25].

Beide Systeme bieten in vielerlei Hinsicht vergleichbare Möglichkeiten – auch TETRAPOL beherrscht neben allen gängigen Features älterer oder einfacherer Systeme Punkt-zu-Mehrpunkt-Verbindungen, Peer-to-Peer-Kommunikation, Infrastrukturmodi, den Einsatz von Endgeräten als Relais sowie (deutlich langsamere) Datenverbindungen unter Berücksichtigung hoher Anforderungen an Integrität, Vertraulichkeit und Abhörsicherheit der

ausgetauschten Informationen. Im Gegensatz zu TETRA unterstützen alle Endgeräte von Haus aus auch verschlüsselte Modi – auch, weil sich das System noch expliziter an zahlungskräftige, professionelle Anwender mit einem entsprechend hohen Sicherheitsbedürfnis richtet.

Trotz vieler Ähnlichkeiten sind beide Lösungen technisch inkompatibel zueinander. TETRAPOL nutzt im Gegensatz zu TETRA Frequenzmultiplex, Vollduplexverbindungen werden durch Zuweisungen zweier verschiedener Frequenzen für beide Kommunikationspartner aufgebaut (Tetra: Zeitmultiplex auf gleicher Frequenz). Auf einem Kontrollkanal sendet die Basisstation (so vorhanden) beständig Steuerdaten des Systems an die mobilen Einheiten – trotz dessen und des zellbasierten Aufbaus ist ein Roaming oder ein Handover für die Endgeräte nicht vorgesehen, ein Zellwechsel erfordert einen Neuaufbau aller Verbindungen. Als Frequenzen kommen wiederum aus dem ehemaligen professionellen Analogfunk umgewidmete Spektren ähnlich wie bei TETRA zum Einsatz, obwohl prinzipiell das gesamte Band von 70 bis 520 MHz in Frage käme [20, Abschnitt „Funktionsweise“]. Der Kanalabstand beträgt 12,5 kHz, die Übertragungsgeschwindigkeit erreicht maximal 7,2 KBit/s Netto (7,6 kBit/s Brutto, für Sprache 6 kBit/s). Die Reichweite soll gegenüber TETRA bei gleicher Sendeleistung deutlich höher sein (fast doppelt so hoch [26]), die Rufaufbauzeit ist mit 300ms hingegen deutlich kürzer [27, Abschnitt „Technik“], [28]. Eine direkte Gegenüberstellung beider Systeme bieten auch die Dokumente [34, S. 7, Tab. 1] und [35, S. 63, Tabelle].

Eine Zusammenschaltung beider Systeme über eine Bridge (i.S.d. Netzwerktechnik) wäre nach Überlegungen verschiedener Stellen durchaus möglich, wenngleich dadurch etliche Leistungsmerkmale verloren gehen würden. Konkret hat dieses Vorgehen EADS bereits vor sieben Jahren demonstriert [29], sogar im gleichzeitigen Zusammenspiel mit älteren Analogfunksystemen. Detaillierte Berichte zum damaligen Versuchsablauf und -ergebnis scheinen aber nicht veröffentlicht worden zu sein.

Heute wird TETRAPOL neben den Sicherheitskräften zahlreicher Nationen unter anderem auch von der Bundeswehr [30, 31], am Frankfurter Flughafen (Fraport) und in Berlin-Tegel sowie bei namhaften deutschen Autobauern eingesetzt [32]. In der Schweiz konnte es sich als BOS-Funksystem etablieren [33].

2.4.2.6. GSM

Wie bereits eingangs unter 2.4.2.1 beschrieben findet GSM bei Verkehrsunternehmen bisher kaum bis gar keine Verwendung als Betriebsfunkmittel. Einzige Ausnahme bildet hierbei der Bereich des Schienenfernverkehrs, für den mit GSM-R eine europaweit zum Einsatz kommende und für den weltweiten Einsatz ausgelegte Technologie geschaffen wurde, die jedoch in einigen wesentlichen Punkten vom klassischen GSM abweicht und die selbst für

größere regional agierende Einzelunternehmen aus wirtschaftlichen Gründen auch nicht zur Verfügung steht.

GSM zeichnet sich im Gegensatz zu allen bisher behandelten Systemen vor allem dadurch aus, dass dafür praktisch überall bereits eine Infrastruktur besteht. Die weißen Flecken auf der Karte der GSM-Netzabdeckung in Deutschland sind in den letzten Jahren kontinuierlich kleiner geworden, mindestens eines der vier großen Netze (D1/Telekom), D2/Vodafone, E1/E-Plus und E2/O₂) ist an beinahe jedem Ort im Freien empfangbar. Damit ist auch jederzeit eine Sprechverbindung zu beliebigen Gegenstellen in den weltweiten Telefonnetzen möglich. Ausnahmen bilden hierbei lediglich durch ihre Lage stark beeinträchtigte Orte wie enge Bergtäler, dicht bewaldetes Hügelland sowie einige sehr dünn besiedelte Regionen in den großen Flächenländern, für die ohne gesonderte Infrastruktur aber auch mit sämtlichen anderen Funktechnologien erhebliche Abdeckungsprobleme entstehen können.

Technisch bietet GSM mit dem fast überall verfügbaren Paketübertragungsdienst GPRS eine Möglichkeit zum Datentransfer zwischen 9,05 und 53,6 kBit/s, an sehr vielen Standorten mit EDGE sogar bis zu 220 kBit/s. Weitergehende, aber nicht mehr direkt auf GSM aufbauende Standards wie UMTS bzw. dessen Verbesserungen HSPA(+) und LTE bieten Bandbreiten bis in den hohen Megabit-Bereich hinein, sind allerdings nicht immer auch flächendeckend verfügbar. Einfache GSM-Endgeräte sind in Form von Mobiltelefonen sowie auch fest in beliebige Geräte integrierbaren Einheiten (GSM-/GPRS-Modems) bereits zu Preisen verfügbar, die keine der Konkurrenztechnologien halten kann und gelten aufgrund der hohen und langjährigen Verbeitung von GSM-Netzen gemeinhin als ausgereift und erprobt.

Trotz dieser augenscheinlichen Überlegenheit konnte sich GSM bisher kaum für innerbetriebliche Anwendungen wie den ÖPNV-Betriebsfunk durchsetzen. Dies liegt zum einen in den beschränkten Nutzungsmöglichkeiten begründet: Die lange Rufaufbauzeit – der klassischen Telefon-Wählvorgang – macht spontane, kurze Mitteilungen unmöglich. Standardfunktionen anderer Funksysteme wie Gruppen- und Rundrufe oder Direktkommunikation zwischen Endgeräten sind nicht vorhanden. Zum anderen bringt die Nutzung eines öffentlichen GSM-Netzes erhebliche technische Einschränkungen mit sich: Einflussnahmen auf die Netzabdeckung oder die Betriebsstabilität sind nicht möglich, man ist auf den Betreiber des GSM-Netzes angewiesen. Es gibt keine Verfügbarkeitsgarantie oder garantierte Mindestleistungen bzgl. der Anzahl an Sprachkanälen und verfügbaren Datenraten. Netzausfälle können gerade in starken Belastungssituationen wie Großveranstaltungen, an denen der ÖPNV besonders gefragt ist, sogar vermehrt auftreten und somit den Betriebsablauf maßgeblich behindern. Außerdem hat der Nutzer kaum Kontrolle über die Sicherheit und Vertraulichkeit der Kommunikation: GSM-Netze können inzwischen auch von Dritten relativ leicht abgehört oder manipuliert werden, was gegenüber Betrieben Möglichkeiten für gezielte Angriffe bietet. Zuletzt erscheint mit der Nutzung

von GSM auch noch ein betriebswirtschaftliches Problem: Die Nutzung muss mit dem Netzbetreiber abgerechnet werden. Pauschalentgelte müssen bei den durch die Vielzahl an Teilnehmereinheiten (mind. eine Einheit pro Fahrzeug oder fahrzeugführendem Mitarbeiter zzgl. Außenpersonal, Zentralen und Leitstellen des Betriebes) hohen möglichen Maximal-Lasten entsprechend hoch angesetzt werden, nutzungsgerechte Abrechnungen hingegen stellen eine betriebswirtschaftlich schwer zu kalkulierende Größe dar und erfordern ein gegebenenfalls nicht erwünschtes Speichern aller Einzelverbindungsdaten seitens des Netzbetreibers, um die abgerechnete Nutzung auch nachweisen zu können. Der Aufbau eines komplett eigenen GSM-Netzes durch ein Unternehmen ist technisch zwar ohne weiteres möglich, wirtschaftlich aufgrund der sehr hohen Investitionen in die Infrastruktur aber selbst von größeren Betrieben kaum zu leisten.

Interessant sind GSM-Netze daher nicht als Ersatz für den klassischen Betriebsfunk, wohl aber für eventuelle Zusatzdienste, welche von diesem völlig unabhängig und in großen Gebieten realisiert werden sollen. Ein Beispiel dafür ist das im Rahmen dieser Arbeit vorgestellte Ortungsmodul: Es soll im Fuhrpark beliebiger Projektpartner eingesetzt werden können, darf daher also nicht von deren jeweiligem Datenfunk abhängig sein. Gleichzeitig muss die technische Lösung für den Datenaustausch äußerst günstig in Anschaffung und Unterhalt bleiben, um das Gerät in großen Stückzahlen einsetzen zu können. Diese Bedingungen sind mit GSM gut erfüllbar: Die je Zeiteinheit zu übertragenden Datenmengen für das Modul sind im Betrieb konstant und sehr gut vorherberechenbar, benötigen aber nur eine geringe Bandbreite. Damit wäre bereits die Nutzung der günstigsten Standardtarife öffentlicher Anbieter möglich, bei hinreichend hohen Stückzahlen kommen jedoch auch Einzelvereinbarungen mit einem der Provider in Frage. Das Modul ist in seinen jetzigen Nutzungsszenarien nicht in echtzeit- oder anwendungskritische Prozesse einbezogen und benötigt keine garantierten Verfügbarkeiten – gelegentliche Netzausfälle oder Einschränkungen, wie sie im Mobilfunknetz auftreten, sind daher durchaus verkraftbar. Aus diesen Gründen wurde das nachfolgend beschriebene Gerät bereits von vornherein zur Verwendung in und mit GSM-Netzen konzipiert.

2.5. Konzept des Ortungsmoduls

2.5.1. Grundanforderungen und Einsatzszenario

Das Ortungsmodul soll als selbstständige Komponente in beliebigen bewegten Objekten zum Einsatz kommen können. *Objekte* meint in diesem Zusammenhang vorrangig landgebundene Fahrzeuge, aber auch der Einsatz zur Verfolgung von Frachtgut oder zur Lokalisierung von Binnenschiffen ist denkbar. Die Möglichkeit Informationen zu erfassen, die über die Position und Bewegung hinausgehen, soll nicht ausgeschlossen werden, darf die

Flexibilität des Gerätes jedoch nicht einschränken. Dies erfordert insbesondere, dass sich das System konzeptuell an verschiedene Szenarien anpassen lässt, woraus sich folgende Grundanforderungen ergeben:

- minimale Anforderungen an die Einsatzumgebung
- Nutzung eines weiten Bereiches verschiedener elektrischer Energiequellen
- Integration aller benötigten Sensorik und Kommunikationsmittel in das Gerät
- geringstmögliche Vernetzung mit dem Objekt zur Herstellung der Grundfunktionen
- Möglichkeit zur Vernetzung mit dem Objekt zur Nutzung zusätzlicher Informationsquellen
- hinreichende Auslegung des Gerätes auf Robustheit und Fehlertoleranz zur wartungsfreien Absolvierung des jeweiligen Einsatzes
- Optimierung auf niedrige Kosten trotz überschaubarer Stückzahlen bei Sicherstellung der mittelfristigen Verfügbarkeit verwendeter oder alternativer Komponenten
- Offenheit des Systems gegenüber Weiterentwicklungen und Veränderungen für zukünftige Nutzungsszenarien

Das Hauptaugenmerk der ersten Entwicklungsstufe liegt auf dem Einsatz einer kleinen Serie dieser Geräte in öffentlichen Verkehrsmitteln der Braunschweiger Verkehrs-AG. Sie sollen als fest in den Fahrzeugen installierte Systeme ohne Interaktion mit der Fahrzeugbesatzung kontinuierlich ihre Position an einen zentralen zur AIM gehörenden Webservice übertragen. Im Sinne der späteren Ausweitung dieses Einsatzszenarios nutzt das System lediglich eine vorhandene Bordenergieversorgung der Busse und Straßenbahnen, bringt jedoch alle benötigten Sensoren und Kommunikationsmittel selbst mit und erfordert nur einen geringen Einbauaufwand, welcher nach kurzer Einweisung auch von einfachem Werkstattpersonal geleistet werden kann. Die Besatzung muss mit dem Modul nicht interagieren.

2.5.2. Prinzipieller Aufbau

Das System besteht aus vier wesentlichen Komponenten:

- zentrale Steuerung in Form eines Mikrocontrollers
- Positionserfassung per GPS-Empfänger
- Datenübertragung mittels GPRS-Modem

- Energieversorgung

Der Mikrocontroller erhält Positionsdaten vom GPS-Empfänger, prüft diese auf Gültigkeit und Aktualität und steuert unter Benutzung des GPRS-Modems die Datenübermittlung an den Webserver. Die Energieversorgung wird von allen drei vorgenannten Teilmodulen in unterschiedlichem Ausmaß beansprucht. Dieser Aufbau ist auch in Abb. 2.5 dargestellt.

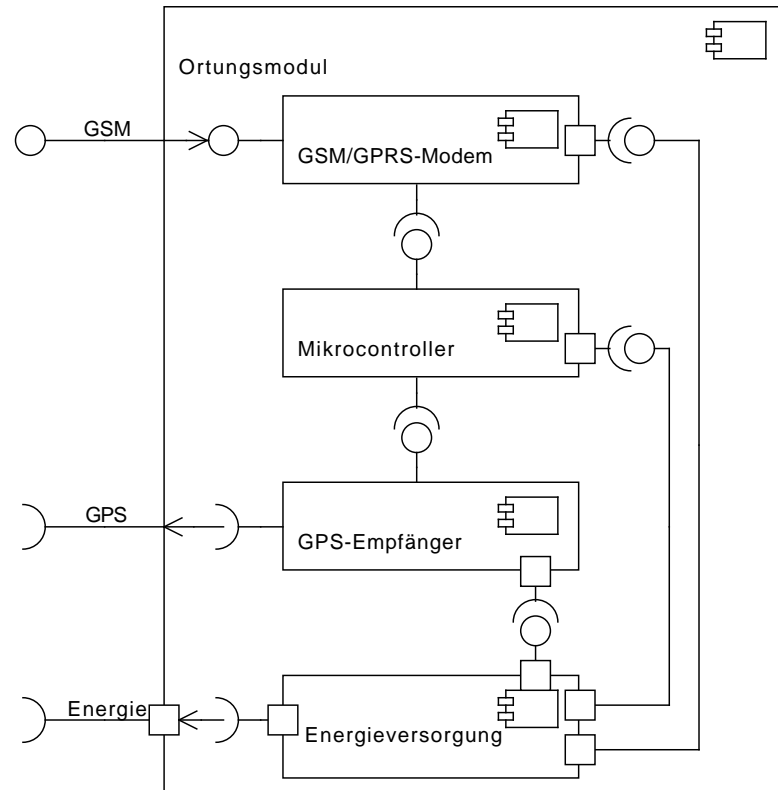


Abbildung 2.5.: Komponentendiagramm des Ortungsmoduls

2.6. Bestehende vergleichbare Systeme

Viele einfache Mobiltelefone stellen bereits seit Einführung der D-Netze ihre internen Modemfunktionen über eine serielle Schnittstelle nach außen zur Verfügung. Fernwirkungssysteme auf dieser Basis existieren daher bereits seit etwa 20 Jahren. Mit dem Aufkommen immer günstigerer GPS-Empfänger etablierte sich mit der Jahrtausendwende auch zunehmend die Idee, diese Systeme zu kombinieren und somit die Position entfernter Objekte zu bestimmen und zu verfolgen. Die Mobiltelefone wurden für professionelle Anwendungen durch spezialisierte GSM-Bausteine ersetzt, inzwischen sind sogar kombinierte GPS/GSM-*SoCs* mit integriertem Mikrocontroller erhältlich (System on Chip). Neben dieser Basisfunktionalität bieten zahlreiche Hersteller auch darüber hinausgehende Möglichkeiten an, von der automatischen Alarmierung eines Autobesitzers im Diebstahlfalle über die Fernbetätigung von Standheizung und Klimaanlage bis hin zur Tür- und Kofferraumüberwachung für Taxen und Kurierdienste. Der Einsatz solcher Systeme im Flottenmanagement größerer Unternehmen ist längst Standard, die Nutzung des GSM-Netzes zur Übertragung von Positionsinformationen oder zur direkten Einwirkung auf technische Systeme an sich also keine Neuerung. Selbst einige Open-Source-Projekte widmen sich in jüngerer Zeit diesem Konzept, womit durchaus Grundlagen für die Weiterentwicklung und Anpassung bestehender Systeme gegeben sind.

Die Motivation zur kompletten Neuentwicklung eines solchen Ortungsmoduls liegt in den spezifischen Anforderungen des AIM-Umfeldes: Das Gerät muss bei minimierten Kosten hohen Anforderungen an seine Einsatztauglichkeit gerecht werden und gleichzeitig im Gegensatz zu bestehenden Lösungen sehr flexibel an verschiedene Einsatzszenarien anpassbar sein. Insbesondere die Sicherstellung der Funktion unter widrigen Bedingungen und die Gewinnung hochwertiger Daten für die Forschung erfordern hier ein teilweise anderes Vorgehen als bei den marktüblichen Produkten – nicht die Gewinnung der einzelnen Positionsinformation steht im Vordergrund, sondern die Generierung kontinuierlicher und zuverlässiger Aufzeichnungen zur Fusion mit weiteren Datenquellen. Die Unabhängigkeit von anderen Projekten und die damit gegebene Anpassbarkeit und langfristige Verfügbarkeit des Ortungsmoduls lassen den im Vergleich zu anderen Teilprojekten überschaubaren Aufwand für diese Eigenentwicklung vertretbar erscheinen.

3. Hardware

3.1. Anforderungen, Auswahlkriterien und -verfahren

Gemäß den Feststellungen aus Kapitel 2.5 besteht das Gerät aus vier wesentlichen Teilsystemen. Die zentralen Aussagen des Kapitels 1.2 schränken die Auswahl der einzusetzenden Technologien klar ein, so dass im Wesentlichen die Entscheidung für ein konkretes Produkt zu treffen bleibt.

Die *Anforderungen* ergeben sich für jedes Teilsystem u.a. aus Kapitel 2.5.1. Hinzu kommt eine Forderung des DLR, den ersten Prototypen auf Basis der *Arduino-Plattform* zu realisieren (siehe 3.2 für eine kurze Erläuterung).

Zur bestmöglichen Erfüllung der aufgestellten Anforderungen wurden folgende *Auswahlkriterien* definiert:

- Kompatibilität mit der Arduino-Plattform
- Modularität: jede Komponente soll für sich abgeschlossen funktionsfähig sein
- Robustheit und Langlebigkeit
- niedrige Kosten
- Offenheit des Systems in Hard- und Software

Das sich daraus ergebende *Auswahlverfahren* jeder Komponente besteht aus 4 Stufen:

1. Erfassung des Angebotes
2. Beschränkung auf Komponenten, welche für die Arduino-Plattform geeignet sind oder über eigene Anpassungen kompatibel gemacht werden können
3. Prüfung des Vorhandenseins offener Software-Beispiele und technischer Unterlagen (Schalt- und Layoutpläne)
4. Sortierung der verbliebenen Ergebnisse nach Preis und Grad der Erfüllung der Grundanforderungen

Im letzten Schritt ließ sich nicht immer ein eindeutig am besten geeignetes Produkt ermitteln, so dass im Folgenden teils mehrere Komponenten verschiedener Anbieter evaluiert wurden.

3.2. Auswahl des Mikrocontrollers

Der erste Prototyp sollte nach einer Anforderung des DLR auf Basis der Arduino-Plattform realisiert werden, welche dort auch in anderen Zusammenhängen zum Einsatz kommt. Dabei handelt es sich um ein Open-Source-Projekt mit dem Ziel der Förderung des Einsatzes von Mikrocontrollern nicht nur im technischen Umfeld, sondern auch für Künstler, Designer, Hobbyisten und jeden, der interaktive Objekte oder Umgebungen erschaffen möchte [36]. Die Arduino-Entwickler stellen dazu neben einer Auswahl standardisierter Hardware für diese auch eine Art einfaches Betriebssystem zur Verfügung. Dieses definiert einen Abstraktionslayer für die Hardware-Peripherie des Mikrocontrollers sowie viele Basisfunktionen, was die Entwicklung von Anwendungen beschleunigen soll.

Derzeit basieren alle Boards des Arduino-Projektes auf Mikrocontrollern der *AVR AT-Mega*-Familie des Herstellers Atmel. Eine Übersicht über die verfügbare Hardware bietet [37]. Zu Projektbeginn waren die Mikrocontrollerboards *Arduino Uno R2* sowie das leistungsfähigere *Arduino Mega 2560* aktuell. Daneben gab es eine Reihe von Boards mit weitergehenden Features wie Ethernet-Anbindung oder einem integrierten SD-Karten-Slot – diese kamen primär aus Kostengründen nicht in Frage. Einen Vergleich der wichtigsten Parameter dieser beiden Basismodule bietet die Tabelle 3.1, eine vollständige Auflistung aller betrachteten Controllerboards und ihrer Eigenschaften findet sich in Anhang A, Seite A-2 ff.

Tabelle 3.1.: Vergleich von Arduino Uno und Arduino Mega 2560

	Controller	SRAM	Flash	Serielle Schnittstellen
Arduino Uno	ATmega 328	2 kB	32 kB	1 (1 belegt)
Arduino Mega 2560	ATMega 2560	8 kB	256 kB	4 (1 belegt)

Diese Parameter sind für die Entwicklung des vorliegenden Projektes von entscheidender Bedeutung. Im SRAM müssen insbesondere große Pufferspeicher angelegt sowie Instanzen der Treiberobjekte für einzelne Komponenten erzeugt werden. Der Flash dient als Programmspeicher – er limitiert zunächst einmal die maximale Größe, die das Kompilat aus Betriebssystem und Anwenderprogramm einnehmen darf. Zusätzlich ermöglicht ein größerer Flash die Auslagerung selten genutzter statischer Objekte aus dem RAM heraus, was diesen signifikant entlasten kann (etwa Strings für Debugmeldungen). Die Anzahl serieller Schnittstellen ist für die Anbindung der übrigen Komponenten wichtig: Jeweils eine ist für die USB-Verbindung der Boards zum Entwicklungsrechner reserviert, zusätzliche Komponenten werden oft über separate serielle Schnittstellen angebunden. Sind keine weiteren als Peripheriemodul des Mikrocontrollers verfügbar, so müssen diese aufwändig über ein Software-Polling implementiert werden, was die Programmausführung enorm verlangsamt.

Obwohl nach der Auswahl der weiteren Komponenten bereits absehbar war, dass das Arduino Uno in all diesen Belangen an seine Grenzen kommen würde, wurde die Entwicklung zunächst auf dem kleineren Modell begonnen: Ein Exemplar dieses Boards war noch aus einem anderen Projekt verfügbar. Mit Beginn der Arbeit an der GSM-Anbindung wurde die Software auf das größere Modell Mega 2560 portiert.

3.3. Auswahl des GPS-Moduls

3.3.1. Vorbemerkung

Die Funktionsweise des hinter GPS stehenden Satellitensystems, die Theorie der Ortung auf Basis von Laufzeitunterschieden mehrerer Satellitensignale, der Umgang mit mehreren Zeitinformationen sowie die mathematischen Hintergründe der von einem GPS-Empfänger durchgeführten Berechnungen sind wichtig zum Verständnis der Funktionsweise GPS-basierter Ortung. Zur tiefergehenden Beschäftigung mit der Materie ist dieses Wissen essenziell, zumal die Berücksichtigung wichtiger Faktoren in der Anwendung solcher Systeme ohne den entsprechenden Hintergrund kaum möglich ist.

Die dazu minimal notwendigen Erläuterungen würden den Rahmen dieser Arbeit jedoch bei weitem überschreiten, weswegen ausdrücklich auf die entsprechende Literatur verwiesen sei ([45], [47], [50, Kapitel 4.7], [54]). Nachfolgend werden lediglich die für die praktische Anwendung zwingend erforderlichen technischen Aspekte von GPS-Empfängern dargestellt, zum Verständnis derselben wird die Lektüre zumindest der einführenden Kapitel der genannten Werke dringend empfohlen.

3.3.2. Aufbau eines GPS-Empfängers

Zur Auswahl eines geeigneten GPS-Modules ist es notwendig, die dahinter stehende Hardware zu betrachten. Als *Modul* bezeichnet man in diesem Zusammenhang ein vollständiges System, welches verschiedene Hardwarekomponenten so miteinander integriert, dass es GPS-Signale empfangen, verarbeiten, in Positionsinformationen umsetzen und diese ausgeben kann. Die notwendigen Hardwarekomponenten umfassen im Wesentlichen den oder die Verarbeitungschip(s), Spannungs- und Signalaufbereitungssysteme, ein spezialisiertes Platinendesign mit genau abgestimmten Masseflächen und Resonanzeigenschaften sowie die zugehörigen Konnektoren zur Integration dieses Systems in bestehende Anwendungen. Eine zugehörige Antenne wird nur dann als Teil eines GPS-Modules betrachtet, wenn diese fest mit diesem verbunden ist (etwa auf der Platine verlötet), separat anschließbare Antennen bilden hingegen einen eigenen Begriff. Eine sehr gute vertiefende Darstellung zu diesem Thema bietet [45, Kapitel 4]

Im Bereich mobiler GPS-Empfänger finden zumeist sogenannte Patchantennen aus Keramik Verwendung. Diese bestehen äußerlich aus einem quadratischen Keramikblock (Schutz und Isolation), innerlich jedoch aus unzähligen auf einem Substrat aufgetragenen Metallplättchen, den namensgebenden Patches. Je nach Bauweise werden diese Patches auf verschiedene Arten zu Arrays gekoppelt und auf einen gemeinsamen Fußpunkt der Antenne geführt. Die Beschaffenheit und Qualität dieses inneren Aufbaus beeinflusst entscheidend die Breitbandigkeit, den Gewinn sowie die Richtwirkung der Antenne, was in der Gesamtheit ihre Empfangsleistung ausmacht. GPS-Antennen werden bei der Verwendung in mobilen Geräten oft direkt auf einer Platine aufgelötet, die auch den Chipsatz und weitere Elektronik trägt. Für größere Anlagen wie Fahrzeuge oder bei professionellen Anwendungen kommen von der eigentlichen Elektronik getrennte Antennen zum Einsatz, welche möglichst günstig positioniert und über Koaxialleiter mit dem Chipsatz verbunden werden können.

Die eigentlichen Empfänger-Chips bestehen intern zumeist aus zwei getrennten Teilen – einem Hochfrequenzbereich (HF) und einem Verarbeitungsbereich (NF, Niederfrequenz). Heute sind diese je nach Herstellungsverfahren auf einem einzigen Die (Siliziumplättchen mit Schaltkreisen) oder zwei miteinander gekoppelten Dies innerhalb eines Packages (Plastikgehäuses mit Anschlusspins) untergebracht. Früher bestanden diese Empfänger aus technologischen Gründen noch aus zwei vollständig getrennten *ICs* (Integrated Circuits, integrierte Schaltkreise), welche immer im Verbund verkauft und eingesetzt wurden – sie werden daher bis heute oft noch als *Chipsatz* anstatt Chip bezeichnet. Aufgabe des HF-Bereiches ist die Umsetzung des analogen Antennensignals in ein Digitalsignal, welches vom nachfolgenden NF-Bereich verarbeitet werden kann. Der NF-Bereich besteht primär aus vielen kleinen Recheneinheiten, den sogenannten Kanälen, welche für die im Abschnitt 3.3.3 beschriebenen Aufgaben zuständig sind. Zusätzlich implementiert der NF-Bereich eine oder mehrere Schnittstellen zur Ausgabe der gewonnenen Informationen und je nach Modell weitergehende Features wie einen Speicher zum Loggen von Wegpunkten, eine interne Spannungsaufbereitung oder Kalenderfunktionen. Einige moderne Systeme benötigen daher neben dem Empfänger-Chip nur noch eine Trägerplatine mit Konnektoren für die Antenne, die Energiezufuhr sowie die Datenausgabe, um ein vollständiges GPS-Modul zu bilden.

3.3.3. Arbeitsweise eines GPS-Empfängers

Das GPS-Satellitensignal besteht aus einer Reihe von digitalen Informationen, welche codiert auf der GPS-Trägerfrequenz von 1575,42 MHz aufmoduliert werden (nur das zivile L1-Signal). Bei GPS kommt das sogenannte Codemultiplexverfahren (CDMA) zum Einsatz, welches es ermöglicht, mehrere voneinander unabhängige Signale auf nur einer einzigen Trägerfrequenz zu senden – eines durch jeden Satelliten. In jedem dieser Einzelsignale sind unter anderem die genaue lokale Uhrzeit des jeweiligen Satelliten sowie die Parameter seiner genauen Bahndaten (die sogenannten Ephemeriden) und – in einem langsameren Sendezyklus – die weniger exakten Bahninformationen aller Satelliten, der Almanach, enthalten. Ein GPS-Empfänger empfängt daher über seine Antenne stets die Signale mehrerer Satelliten gleichzeitig und steht vor der Aufgabe, diese wieder in ihre Einzelinformationen zu zerlegen. Die HF-Einheit des Empfänger-Chipsatzes kann dabei nur das analoge Satellitensignal abtasten und aufbereiten, um ein für die NF-Einheit verwertbares Digitalsignal zu generieren. Eine weit ausführlichere Darstellung des Aufbaus des GPS-Signals sowie der damit geleisteten Informationsübertragung findet sich in [45, Kapitel 3] sowie kompakter in [47, Kapitel 4.2].

Die NF-Einheit verfolgt gleichzeitig den sich zyklisch wiederholenden Datenstrom mehrerer Satelliten. Dazu muss sie diese Datenströme voneinander trennen, was aufgrund des verwendeten CDMA die Berechnung von Kreuzkorrelationen zwischen den einzelnen Signalen erfordert. Dies ist ein äußerst rechenintensiver und komplexer Vorgang, der sich mit herkömmlichen Prozessoren nicht bewältigen lässt. Die Berechnungen erfolgen daher parallel in den vielen einzelnen Kanälen einer solchen NF-Einheit. Aus den einzelnen Datenströmen gewinnt der Empfänger dann die Parameter zur Berechnung der Ephemeriden des jeweiligen Satelliten – sie geben Auskunft über seine Satellitenbahn, womit die Bestimmung des Satellitenortes zum Sendezeitpunkt des Signals möglich wird. Ihre Gültigkeit ist analog der Zeit, die ein Satellit maximal am Himmel steht, auf etwa 4 bis 6 Stunden begrenzt. Durch Messung des Laufzeitunterschiedes zwischen mehreren verschiedenen Signalen lässt sich so die Entfernung des Empfängers von den einzelnen Satelliten gewinnen – zusammen mit der Ortsinformation lässt sich daraus seine Position bestimmen. Die Effizienz des verwendeten Verfahrens zur Berechnung der Kreuzkorrelationen sowie die Anzahl der Kanäle sind entscheidend für die Leistungen des Chipsatzes, weswegen die meisten Unternehmen diese Kerntechnologie von einem der großen Anbieter lizenzieren und erst diese *GPS Engine* genannte Komponente mit eigener Technologie zu einem Chipsatz oder GPS-Modul erweitern.

Die Empfänger legen Vorhersagen über die längerfristig gültigen ungefähren Bahndaten der Satelliten an – den Almanach. Dessen Informationen bleiben etwa 3 Monate lang gültig. Wird ein GPS-Empfänger eingeschaltet, ermittelt er ausgehend von seiner letzten Position sowie diesem Almanach die am Himmel zu erwartenden Satelliten, was die Zahl

möglicher Korrelationen deutlich einschränkt und so einen schnelleren Ortung ermöglicht. Die Zeit vom Start eines Empfängers bis zur Ermittlung der ersten gültigen Position (erster Fix) nennt man *TTFF* (Time to first Fix). Bei einem *heißen* Start – einem Neustart des Empfängers ohne Unterbrechung der Energiezufuhr – sind die Ephemeridendaten noch vorhanden und gültig, wodurch die erste Positionsbestimmung bereits nach Bruchteilen einer Sekunde erfolgen kann. Bei einem *warmen* Start sind diese Daten bereits veraltet, der noch vorhandene Almanach kann die Zeit bis zum ersten Fix aber dennoch signifikant senken (wenige zehn Sekunden bis einige Minuten.) Liegt kein gültiger Almanach vor, so spricht man von einem *kalten* Start. Dieser tritt auch dann ein, wenn der Empfänger seit seiner letzten Ortung um mehr als 300 km bewegt wurde, da die erwarteten Satelliten dann nicht mehr mit den tatsächlich sichtbaren übereinstimmen müssen – in diesem Falle muss bei älteren Empfängern der gesamte Almanach neu herunter geladen werden, bevor ein erster Fix erreicht werden kann. Dies dauert unabhängig vom jeweiligen Empfänger mindestens die 12,5 Minuten eines vollständigen Almanach-Sendezyklus, aufgrund von Übertragungsfehlern oft auch länger. Neuere Empfänger verfügen hingegen über ein so genanntes *All-in-View-Tracking*, d.h., sie sind in der Lage, die Datenströme *aller* für sie am Himmel sichtbaren Satelliten gleichzeitig zu verfolgen. Damit können sie i.d.R. bereits hinreichend viele Ephemeriden und damit auch einen ersten Fix erhalten, lange bevor der Almanach aktualisiert wurde.

Die durchschnittlich benötigte Zeit für diese drei Startvorgänge bildet neben der Anzahl der Kanäle und der Empfangsempfindlichkeit ein wesentliches Kriterium zur Beurteilung der Qualität von GPS-Empfängern. Zur Reduzierung dieser Zeitspannen können vorausberechnete Almanachdaten über diverse Online- und Mobilfunkdienste bezogen werden, man spricht dann von *AGPS* (Assisted GPS). Dieses Verfahren wird vor allem von Smartphones und hochwertigeren Tragbaren GPS-Trackern verwendet [45, S. 7 f.].

Für einige Regionen der Erde sind zusätzliche stationäre Satelliten im Einsatz, welche auf derselben Frequenz wie die GPS-Satelliten ein Signal mit regional gültigen Korrekturinformationen senden, welches die Positionsgenauigkeit erheblich zu verbessern hilft. Für Europa heißt dieses System *EGNOS* (European Geostationary Navigation Overlay Service), generell nennt man dieses Verfahren *SBAS* (Satellite Based Augmentation System) [45, S. 8 – 15 sowie Kapitel 6]. Es wird inzwischen auch von vielen preiswerteren Empfängern unterstützt, da sich der technische Mehraufwand zur Nutzung dieser Information in Grenzen hält. Aufgrund der Idee, zusätzliche, regional gültige Korrekturinformationen in die Berechnung der Position mit einzubeziehen, nennen einige Hersteller dieses Verfahren auch *DGPS* (Differential GPS) – es darf jedoch nicht mit dem gleichnamigen Verfahren verwechselt werden, bei dem eine Basisstation an einem exakt eingemessenen Referenzpunkt aufgestellt wird und dort permanent ihre eigene GPS-Position ermittelt. Den Fehler der GPS-Position zu ihrem wahren Standort überträgt diese Basisstation dann zumeist per Funk an eine in der Nähe befindliche mobile Einheit, welche ihre Position entspre-

chend korrigieren kann. Mit diesem Verfahren lassen sich Genauigkeiten von unter 1 cm erreichen, es ist jedoch sehr hochpreisigen Empfängern vorbehalten und erfordert einen entsprechenden infrastrukturellen Aufwand.

3.3.4. Verfügbarkeit und Auswahl der GPS-Module

Es gibt eine breite Palette von GPS-Modulen, welche für die Anbindung an Mikrocontroller wie die ATMegs geeignet sind. Direkt für die Arduino-Plattform entwickelt waren zum Stand November 2011 zwei verschiedene Lösungen: Ein miniaturisiertes GPS-Modul nach XBee-Formfaktor zur Verwendung auf einem sonst für Funkkommunikation gedachten XBee-Shield sowie ein GPS-Shield, der zur Anbindung verschiedener aufsteckbarer GPS-Module geeignet ist. Als *Shield* bezeichnet man Module, welche in Abmessungen und Pinbelegung exakt den Arduino-Basiscontrollern entsprechen oder zumindest mit ihnen kompatibel sind, so dass sie ohne Lötverbindungen direkt auf die Grundplatten aufgesteckt werden können.

Neben diesen Shields existierte eine Reihe von nicht direkt für die Arduinos konzipierten, prinzipiell aber ebenfalls geeigneten Modulen. Die vollständige Auflistung aller betrachteten GPS-Lösungen und ihrer Eigenschaften findet sich in Anhang A, Seite A-6 ff.

Das erste entscheidende Auswahlkriterium ist die Möglichkeit, an das jeweilige Modul eine externe GPS-Antenne anschließen zu können. Diese soll auf dem Fahrzeugdach befestigt werden, um durch die erhöhte Position, die freie Sicht zum Himmel und den verminderten Einfluss anderer Fahrzeugelektronik die Empfangsleistung zu verbessern (metallenes Fahrzeugdach als Schirmungsfläche). Das zweite Kriterium ist der Preis je Modul, wobei dieser im Rahmen des Budgets nicht zur Unterschreitung der erforderlichen Mindestgenauigkeit der Positionsdaten führen darf (Zuordnung Fahrzeug \leftrightarrow Straße mittels Mapmatching muss möglich bleiben). Anhand dieser Kriterien wurden zwei GPS-Module ausgewählt und in einem direkten Vergleich auf ihre Leistungsfähigkeit hin untersucht.

3.3.5. Evaluation ausgewählter GPS-Module

3.3.5.1. Überblick

Das *Venus GPS Board* des Anbieters Sparkfun verwendet den Chipsatz *Venus634FLPx* von SkyTraq. Es bietet die Möglichkeit zum Anschluss einer aktiven oder passiven externen Antenne über einen SMA-Verbinder und stellt die Positionsinformationen in Form eines proprietären, aber dokumentierten binären Protokolls oder wahlweise in Form von NMEA-Datensätzen zur Verfügung. Es wurde primär wegen seines niedrigen Preises und

der hohen Updaterate ausgewählt. Zum Einsatz kam es im Rahmen dieser Arbeit mit einer für den Fahrzeugeinsatz geeigneten passiven Patchantenne, welche in einem wasserdicht vergossenen Plastikgehäuse per Magnetfuß auf dem Fahrzeugdach befestigt und über ein 5 m langes Koaxialkabel angeschlossen werden kann.

Das *MOD-GPS* von Olimex nutzt den verbreiteten SiRF Star III Chipsatz und integriert diesen samt einer Patchantenne vollständig in einem wasserdicht vergossenen Plastikgehäuse. Auch dieses kann magnetisch auf dem Fahrzeugdach befestigt werden, über die rund 3 m lange Zuleitung wird neben der seriellen Datenübertragung auch die Energieversorgung gewährleistet. Es bietet neben dem binären, aber weit verbreiteten SiRF-Protokoll ebenfalls die Ausgabe über NMEA-Datensätze an.

Die (nur einmalig notwendige) Konfiguration beider Empfänger kann über Software der jeweiligen Chipsatz-Hersteller erfolgen, welche auf deren Webseiten für Windows-Betriebssysteme als Download erhältlich ist. Zum Teil existieren jedoch auch freie Alternativen zu diesen Programmen, welche dieselben binären Protokolle implementieren und nutzen.

3.3.5.2. Vergleich der Spezifikationen laut Hersteller

Tabelle 3.2 bietet einen Vergleich der Herstellerangaben zu wichtigen Parametern. Die Werte wurden für die mittlere Versorgungsspannung sowie die Spezifikation der seriellen Schnittstelle direkt überprüft. Die Startzeiten, Updateraten sowie die erreichte Positionsgenauigkeit wurden separat evaluiert, siehe dazu auch Kapitel 3.3.5.3 bis 3.3.5.5.

Das MOD-GPS kann gemäß [39, S. 3] direkt mit den Schnittstellen des Arduino-Basiscontrollers verbunden werden, da es ebenfalls mit CMOS-Pegeln von 5V arbeitet und versorgt wird. Im Widerspruch dazu gibt das Datenblatt [39, S. 11] jedoch auch die Möglichkeit zur Nutzung von 3.3 V an. Tests mit beiden Werten verliefen positiv. Demgegenüber benötigt das Venus-GPS für den Einsatz im ersten Prototypen eine Schaltung zur Pegelanpassung, da es nur für den LVTTTL-Pegel von 3.3 V ausgelegt ist. Diese Einschränkung kann in nachfolgenden Evolutionsstufen des Systems jedoch entfallen, falls die Systemversorgung generell auf 3.3 V abgesenkt wird.

Beide Module verfügen mit ihrer hohen Kanalzahl über ein All-in-View-Tracking und versprechen damit einen Heißstart-Fix binnen einer Sekunde. Die erheblich niedrigere Kaltstartzeit sowie die höhere Updaterate bei geringerem Energieverbrauch lassen das Venus-GPS interessant erscheinen. Diese Werte sowie die Ortungsgenauigkeit beider Module wurden unabhängig von den Herstellerangaben unter realen Bedingungen evaluiert. Die Ergebnisse finden sich in den folgenden Kapiteln.

Tabelle 3.2.: Parameter der verglichenen GPS-Module lt. Datenblättern [38] und [39]

	Venus GPS Board	Olimex MOD-GPS
Chipsatz	Venus634FLPx	SiRF Star III
Kanäle	51 (Signalfindung) + 14 (Tracking)	20 gesamt
Updaterate	bis zu 10 Hz	1 Hz
Empfindlichkeit	-161 dBm	-159 dBm
Kaltstartzeit (TTFF)	29 Sekunden	48 Sekunden
Warmstartzeit (TTFF)	k.A.	38 Sekunden
Heißstartzeit (TTFF)	1 Sekunde	1 Sekunde
Mittlere Genauigkeit	< 2.5 m	5 – 25 m
SBAS-Unterstützung	ja	ja
Versorgungsspannung	2.7 – 3.6 V	3.7 bis 6.0 V
Stromverbrauch (Tracking)	28 mA	75 mA
Unterstützung für Warmstarts	ja (ext. Speisung)	ja (int. Lithiumakku vorh.)
Stützspannungsbereich	1.5 – 6.0 V	3 V
Pegel serielle Schnittstelle	LVTTL (3.3 V)	CMOS (5 V)
Baudrate serielle Schnittstelle	4800 – 115200 Baud	1200 – 57600 Baud

3.3.5.3. Messung der TTFF und Updaterate

Die Almanachdaten beider Module lassen sich über die Software des jeweiligen Herstellers setzen oder entfernen. Nach dem Entfernen der Almanachdaten wurden durch kurzes Unterbrechen der Energiezufuhr Kaltstarts der Module erzwungen. Die Zeit bis zum ersten Fix wurde gemessen und als Kaltstartzeit notiert. Dieser Vorgang wurde dreimal wiederholt.

Anschließend bekamen die Module eine Stunde Zeit zum vollständigen Herunterladen aktueller Almanachdaten, um nicht auf die nur schwer überprüfbaren Daten ausweichen zu müssen, die die jeweilige Software auf die Module laden könnte. Danach wurden sie für mehr als 6 Stunden von ihrer Hauptenergiezufuhr getrennt, um die Ephemeriden altern zu lassen. Beiden Modulen stand hierbei eine Pufferenergiequelle zur Erhaltung der Almanachdaten zur Verfügung.

Beim folgenden Einschaltvorgang wurde die Zeit vom Zuschalten der Spannungsversorgung bis zum Erreichen des ersten Fixes gemessen und als Warmstartzeit notiert. Anschließend wurde die Hauptspannung für jeweils zehn Sekunden aus- und danach wieder eingeschaltet. Die Zeit bis zum ersten Fix wurde als Heißstartzeit notiert. Die Module wurden noch einige Minuten in Betrieb gelassen, um den Pufferspeichern Zeit zum Laden zu geben. Danach wurden sie wieder für über 6 Stunden von Ihrer Hauptenergiezufuhr

getrennt und zwei weitere Messungen nach diesem Schema durchgeführt. Die Messprotokolle finden sich im Anhang B.3, die wesentlichen Ergebnisse sind in der Tabelle 3.3 zusammengefasst.

Tabelle 3.3.: Mittelwerte der gemessenen Startzeiten

Wert	Venus-GPS	MOD-GPS
Kaltstartzeit	30,7 Sekunden	36,0 Sekunden
Warmstartzeit	6,7 Sekunden	30,3 Sekunden
Heißstartzeit	< 1 Sekunde	< 1 Sekunde

Die Ergebnisse beider Module liegen mit diesen Werten durchaus im Bereich hochwertiger moderner Empfänger für den nicht-professionellen Einsatz. Auffällig ist vor allem die sehr kurze Heißstartzeit von weniger als einer Sekunde – ein Feature, welches noch vor einigen Jahren ohne All-in-View-Tracking unmöglich gewesen wäre – sowie die aus dem gleichen Grund mögliche Kaltstartzeit von deutlich unter einer Minute, da der Empfänger nicht mehr den kompletten Almanach herunterladen muss, um einen ersten Fix zu erreichen. Die etwas schlechteren Durchschnittswerte des MOD-GPS sind für den Einsatz im Rahmen dieser Arbeit kaum relevant.

Zur Überprüfung der vom Hersteller angegebenen maximalen Updateraten wurden die Module über die Herstellersoftware entsprechend konfiguriert (10 Hz bzw. 1 Hz). Anschließend wurde deren Ausgabe mitgeschnitten und über die NMEA-Zeitstempel ein 120 Sekunden langes Zeitfenster der Ausgabe ausgewählt. Die Zahl der darin befindlichen GPGLA-Sätze entsprach mit 1173 (Venus-GPS) bzw. 120 (MOD-GPS) annähernd den Erwartungswerten (1200 bzw. 120 Werte). Die Differenz im Falle des Venus-GPS lässt sich mit Übertragungsfehlern erklären, welche durch den Versuchsaufbau begünstigt wurden (siehe auch Anhang B.2). Die in den Datenblättern angegebenen Updateraten können daher als erreicht bezeichnet werden – dies sagt jedoch nichts darüber aus, wie gut ein Empfänger auch einer entsprechend schnellen Fahrzeugbewegung zu folgen vermag.

3.3.5.4. Vergleich der Ortungsgenauigkeit im stationären Einsatz

Zur Ermittlung der Genauigkeit der ausgewählten GPS-Module wurde eine Vergleichsmessung durchgeführt. Ein Teil dieser Vergleichsmessung war das Positionieren der Module auf einem exakt eingemessenen Referenzpunkt, um zu beobachten, ob bzw. wie exakt die ausgegebenen Koordinaten mit den Referenzwerten übereinstimmen. Die Messungen wurden auf dem Gelände des DLR Braunschweig am 13. Oktober 2011 ab 13:55 Uhr UTC durchgeführt. Zum Einsatz kam neben den zu vergleichenden Modulen ein Empfänger der Firma Javad, konkret das Modell Delta mit einer Antenne vom Typ GrAnt. Dieses System ist für professionelle Anwendungen mit entsprechendem Budget ausgelegt und wird

im DLR normalerweise für Vergleichsmessungen oder im Rahmen der Erprobung hochgenauer Ortungsverfahren eingesetzt. Der Delta-Empfänger wurde konfiguriert, nur das GPS-Satellitensystem zu nutzen und die internen Filter zur Stabilisierung der Position zu deaktivieren. Das Venus-GPS nutzte den *Car*-Modus (einzige Alternative: *Pedestrian*), im MOD-GPS lassen sich keine Filter konfigurieren.

Die Empfänger wurden erst nach dem Einschalten zu dem genutzten Referenzpunkt gefahren und erhielten auch dort genug Zeit zum Finden eines ersten Fixes. Die Geräte verblieben rund 20 Minuten am Standort, bevor weitere Messungen folgten. Für die Berechnung der mittleren Position wurden ausschließlich Wegpunkte im Zeitraum von 14:14:00.000 Uhr bis 14:19:00.000 Uhr UTC verwendet, so dass die Module über 15 Minuten Zeit hatten, sich der exakten Position anzunähern. Die NMEA-Daten der Messungen sowie die Dokumente mit den zugehörigen Mittelwertberechnungen befinden sich aufgrund ihrer Größe ausschließlich im elektronischen Anhang dieser Arbeit. Tabelle 3.4 listet die Ergebnisse auf.

Tabelle 3.4.: Mittelwerte der gemessenen Positionen und deren Abweichung vom Referenzpunkt

Empfänger	Breitengrad	Längengrad	Distanz zum Referenzpunkt (m)
(Referenzpunkt)	52,315867648	10,561733418	—
Venus-GPS	52,315861700	10,56176330	1,67
MOD-GPS	52,316201915	10,56120046	46,14
Javad Delta	52,315868946	10,56171528	0,94

Die Position des Referenzpunktes ist in der Länge auf 1,5 cm und in der Breite auf 5 cm genau eingemessen. Die angegebenen Mittelwerte der gemessenen Positionen der Empfänger sind das ungewichtete arithmetische Mittel der jeweiligen Breite bzw. Länge. Die angegebene Distanz zum Referenzpunkt wurde als Abstand zweier Punkte auf einer Kugeloberfläche mit $r = 6371000,8m$ berechnet, was dem mittleren Erdradius gemäß dem auch bei GPS verwendeten WGS84-Referenzellipsoiden entspricht. Die Abweichung gegenüber einer Berechnung unter Verwendung des Äquator- oder Polradius beträgt hierbei weniger als 2 cm / m, weswegen diese sowie die Höhe über NHN nicht berücksichtigt wurden. Die Distanz $d(P_E, P_R)$ ergibt sich daher zu:

$$d(P_E, P_R) = \frac{\pi \cdot r}{180} \cdot \cos^{-1}(\cos(B_E) \cdot \cos(B_R) \cdot \cos(L_E - L_R) + \sin(B_E) \cdot \sin(B_R))$$

mit $P_E(B_E, L_E)$ als Position des Empfängers

und $P_R(B_R, L_R)$ als Referenzpunkt

Der mathematische Vergleich offeriert bereits erhebliche Unterschiede zwischen den verschiedenen Geräten: Das MOD-GPS weicht in seiner mittleren Position um über 46 Meter von der realen Position ab, die anderen beiden Empfänger erreichen Genauigkeiten

im Grenzbereich der Leistungsfähigkeit reiner GPS-Messungen. Ein Vergleich der aufgezeichneten Daten zeigt, dass die sehr viel schlechtere Leistung des MOD-GPS nicht durch die Umgebung beeinträchtigt wurde – es waren konstant 10 oder mehr Satelliten sichtbar. Diese wurden von den Modulen allerdings in unterschiedlichem Ausmaß genutzt – das MOD-GPS hat bereits bei der Empfangsleistung defizite, welche sich bei der Positionsbe-
rechnung und -filterung bemerkbar machen.

Zur weiteren Bewertung der Messergebnisse wurden sämtliche aufgezeichneten Positionen des betrachteten Zeitraumes auf einer Karte grafisch dargestellt. Jedes Datum ist durch einen farbigen Punkt markiert, Linien verbinden die Punkte jedes Empfängers in der Reihenfolge ihrer Aufzeichnung. Der Referenzpunkt ist mit einem hellroten Markierer hervorgehoben. Abbildung 3.1 zeigt das Gelände in etwas größerem Maßstab, Abbildung 3.2 den Bereich um den Referenzpunkt. Zu beachten ist, dass die hinterlegten Luftbilder nicht exakt senkrecht aufgenommen wurden und etwas nach Süden verschoben sind, der Referenzpunkt befindet sich genau auf dem südlichen Bordstein der gezeigten Straße.



Abbildung 3.1.: Übersicht der aufgezeichneten Messpunkte



Abbildung 3.2.: Detailaufnahme des Bereiches um den Referenzpunkt

Interessant ist die weit entfernte nordwestliche Position des MOD-GPS (grün). Diese wurde bis etwa 40 Sekunden vor Ende der Messung gehalten und ist hauptverantwortlich für die große Abweichung des Empfängers vom Zielpunkt. Die Ursache der späten Annäherung an den Referenzpunkt ist nicht bekannt, korrespondiert aber mit dem Verhalten des Javad-Delta-Empfängers (s.u.).

Das Venus-GPS zeigte während der gesamten Messung keinerlei Bewegung. Möglicherweise wurden nach erfolgter Annäherung an die letztendlich gehaltene Position alle kleineren Bewegungen vom internen Filter des Empfängers unterdrückt, welcher auf den Bewegungstyp *Car* eingestellt war. Empfänger für Alltagsanwendungen nutzen oft Filter- und Schwellwertsysteme, um ihr Positionsrauschen zu mindern und fangen z.B. erst bei Erreichen einer Mindestgeschwindigkeit an, ihre Position wieder zu aktualisieren. Das genaue Verhalten der Filter des Venus-GPS ist in den öffentlich verfügbaren Unterlagen von Skytraq nicht dokumentiert.

Der Javad-Empfänger erreichte trotz weitgehend deaktivierter Filtersysteme erwartungsgemäß die beste Genauigkeit. Die nach Südosten weglaufenden Punkte stammen jedoch vom Ende der Messung, sie fallen in dieselben 40 Sekunden, während denen sich das MOD-GPS dem Referenzpunkt annäherte. Dies lässt die Vermutung zu, dass beide Empfänger dem selben Einfluss unterlagen, die verfügbaren Daten geben jedoch keinerlei Aufschluss über die mögliche Ursache.

3.3.5.5. Vergleich der Ortungsgenauigkeit in Bewegung

Nach Abschluss der stationären Messungen wurden die Empfänger gemeinsam auf ihre Genauigkeit bei Bewegung hin untersucht. Der Versuchsaufbau bestand aus einem einfachen zweiachsigen Handwagen ohne Aufbauten, auf dessen Plattform die Empfänger, ein Laptop zur Datenaufzeichnung sowie eine KFZ-Batterie zur Energieversorgung installiert wurden. Die Batterie schränkte die Sicht der Empfänger in Fahrtrichtung um etwa 20° ein, die umliegenden Gebäude teils deutlich mehr.

Da keine eingemessene Referenzstrecke zur Verfügung stand, der die Empfänger hätten exakt folgen können, wurde keine mathematische Auswertung vorgenommen. Die Abbildungen 3.4 bis 3.5 zeigen jedoch die aufgezeichneten Daten aller drei Empfänger, inklusive der Annäherung und stationären Messung am Referenzpunkt.

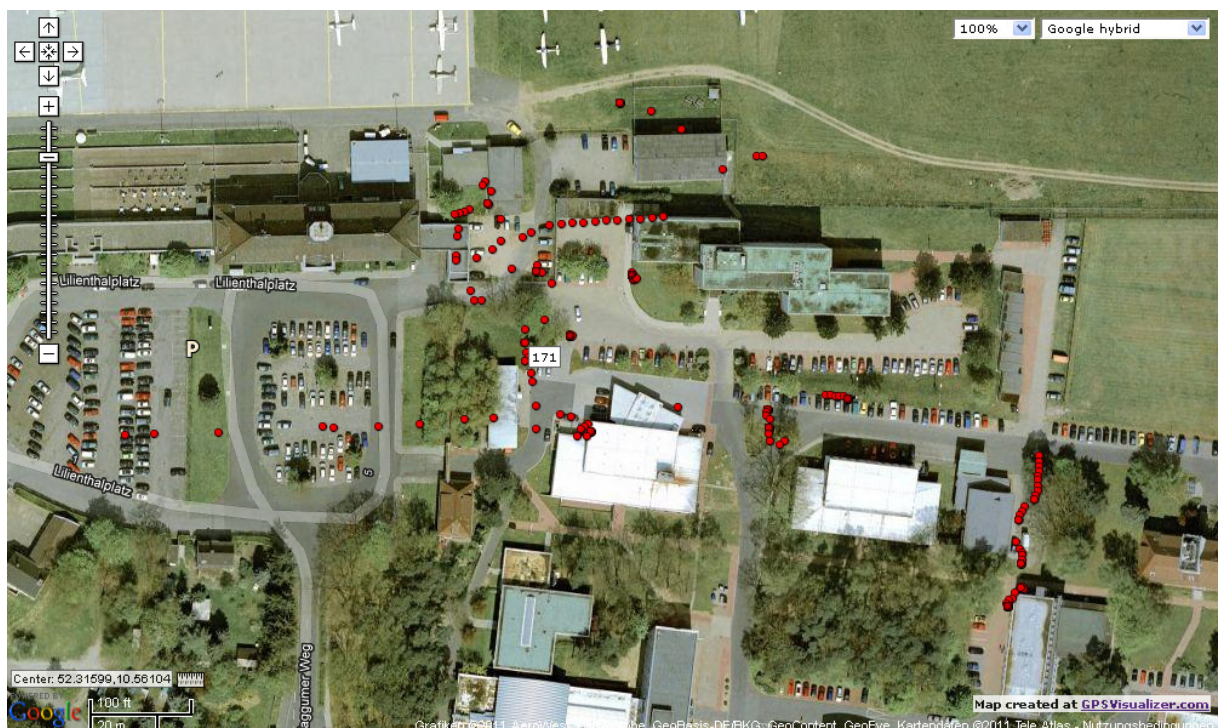


Abbildung 3.3.: Bewegungsspur des MOD-GPS



Abbildung 3.4.: Bewegungsspur des Venus-GPS

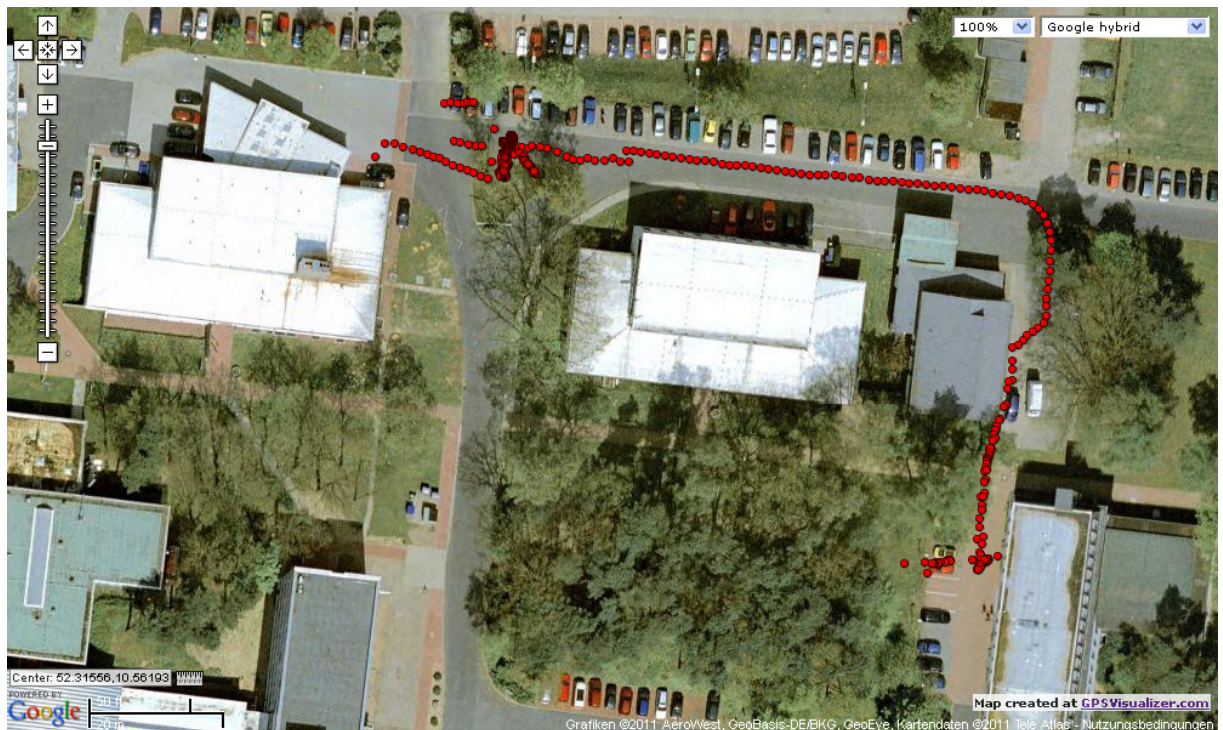


Abbildung 3.5.: Bewegungsspur des Javad-Delta-Empfängers

Die Auswertung der grafischen Darstellung der Bewegungsspuren sowie der NMEA-Datensätze bringt bemerkenswerte wie auch unerwartete Ergebnisse. Die reale Bewegungsspur verläuft vom Referenzpunkt der stationären Messung aus am nördlichen Rand der Straße entlang in östlicher Richtung, bevor ein scharfer Wechsel auf die südliche Straßenseite folgt (zu sehen nur in Abb. 3.4). Der weitere Weg führt östlich bis zum Abzweig einer Straße nach Süden und folgt dieser am westlichen Straßenrand entlang. Nördlich der auf den Luftaufnahmen zu erkennenden parkenden Fahrzeuge in der südlichen Stichstraße geht die Bewegungsspur auf den Fußweg über (zu erkennen in Abb. 3.5), führt über den Querweg nach Süden und dann vor dem großen Bürogebäude entlang. Nördlich des Gebäudeeinganges (erkennbar auf den Fotos als Personengruppe) wurden die Empfänger abgeschaltet. Die NMEA-Datensätze aller drei Empfänger befinden sich auch im elektronischen Anhang dieser Arbeit.

Die Positionsdaten des Olimex MOD-GPS streuen nach dem Einschalten in einem weiten Bereich nordwestlich des Referenzpunktes, bevor sie sich diesem annähern. Die Entfernung beträgt teils deutlich über 100 m, die Sprünge zwischen den einzelnen Werten sind sehr groß. Nach Abschluss der stationären Messungen ortet sich das Modul zwar auf der genutzten Straße, aber deutlich zu weit nördlich. Nach wenigen Metern verliert es den Empfang – womöglich aufgrund des großen, verschattenden Gebäudes im Süden – und ortet sich erst beim Einbiegen in die nach Süden verlaufende Stichstraße erneut. Die Positionen liegen ab Einfahrt in die Straße etwas zu weit nordöstlich, der grobe Verlauf der Bewegungen stimmt jedoch trotz mehrfacher Empfangsverluste. Ab Überquerung des schmaleren Fußweges und im Bereich des größeren Bürogebäudes stimmen die Positionen nicht mehr, sie liegen zeitlich teils mehrere zehn Sekunden vor oder hinter der gelaufenen Strecke und springen und mehrere Meter.

Der Venus-GPS-Empfänger von Sparkfun liefert als ersten Punkt unmittelbar nach dem Einschalten eine Position im Herzen Taiwans – ein Verhalten, welches auch später mehrfach beobachtet wurde und wohl eine Hommage des Herstellers SkyTraq an dessen Heimat darstellt. Der erste Fix wird deutlich zu weit südlich gefunden (zwischen den Gebäuden im Nordwesten und Zentrum des Bildes), nach etwas weniger als zwei Sekunden springt die Position nach Nordwesten und nähert sich von da aus in unter 20 Sekunden bereits auf knapp 2 m an den Referenzpunkt an. Zwei Minuten nach dem Einschalten ist bereits die Position am Referenzpunkt erreicht, welche auch die gesamte stationäre Messung über beibehalten wird. In dem Moment, in dem der Messwagen sich in Bewegung setzt, springt die Ausgabe sogar noch einmal deutlich näher an den Referenzpunkt heran. Das Modul folgt der gefahrenen Route am nördlichen Straßenrand und dem scharfen Wechsel der Straßenseite sehr präzise, beim Einbiegen in die Stichstraße sind die Messwerte jedoch um fast eine ganze Straßenbreite nach Nordosten verschoben. Im Verlauf der Straße nach Süden orientiert sich das Venus-GPS fortwährend zu weit östlich und verpasst damit auch den Wechsel auf den Fußweg nahe an dem Gebäude westlich der Straße – womöglich spielt

hier die Verschattung durch die Bäume und das Gebäude im Süden eine Rolle, die Zahl sichtbarer Satelliten nimmt laut den aufgezeichneten Daten jedoch nicht ab. Südlich der Querung des Fußweges folgt das Modul präzise dem Verlauf der Gebäudewand (das Bildmaterial ist nicht exakt senkrecht aufgenommen und liegt damit etwas zu weit östlich. Beim Abstellen des Wagens hinter dem Windfang des Gebäudes springt die Position etwa 3 m nach Südwesten.

Neben den für den Einsatz im Ortungsmodul zu evaluierenden Modulen wurde auf der Messfahrt auch ein professioneller Delta-Empfänger von Javad mitgeführt. Ziel dessen war es, den Einfluss der Umgebung auf das Satellitensignal möglichst deutlich sichtbar zu machen, weswegen interne Fehlerkorrekturverfahren und Filtersysteme weitgehend deaktiviert wurden. Dessen Messdaten weisen daher im Schnitt auch eine niedrigere Zahl an Satelliten auf, deren Informationen zur Berechnung der Position herangezogen wurden (ohne Filtersysteme können schlechte, aber empfangene Signale nicht mehr verwendet werden). Die ersten Positionen des Empfängers liegen bereits nahe des Einschaltpunktes vor der Fahrzeughalle westlich des Referenzpunktes und nähern sich letzterem noch etwas schneller an als die des Venus-GPS. Während der stationären Messungen unterliegen die Werte fortlaufend dem leichten Rauschen, welches beim Venus-GPS durch die internen Filter unterdrückt wird. Zu Beginn der eigentlichen Messfahrt wird deutlich, dass das große Gebäude südlich der Straße einen signifikanten Einfluss auf die Empfangsqualität hat – das Signal und damit die Anzahl nutzbarer Satelliten verschlechtert sich, der Empfänger ändert seine Position etwas nach Norden. Der Wechsel der Straßenseite wird nicht registriert, das Gerät bleibt mehr als 3 m zu weit nördlich und folgt auch nicht dem wahren Verlauf der Kurve beim Einbiegen in die nach Süden verlaufende Straße. Die Form der aufgezeichneten Kurve sowie der sehr gerade Streckenverlauf vor und hinter dieser legen nahe, dass das Gerät die Werte teilweise interpoliert und stark korrigiert hat – die Untersuchung dieses Verhaltens wurde ein eigenständiges internes Projekt im DLR. Den Wechsel von der Stichstraße auf den Fußweg registriert das Modul wieder präzise und folgt diesem auch korrekt bis an den quer verlaufenden Weg – in diesem Bereich weisen die Werte des Venus-GPS erheblich größere Abweichungen auf. Fortwährende kleine Korrekturen und Richtungswechsel auf dem letzten Teilabschnitt deuten auf ein schlechtes Signal mit vielen Reflexionen hin, begünstigt durch die hohen Objekte in der Umgebung. Auch der Delta-Empfänger wird durch das Abstellen hinter dem Windfang des Bürogebäudes bis zu seiner Abschaltung signifikant gestört.

Das sehr störanfällige Verhalten des Olimex MOD-GPS wurde ob des guten Rufes der SiRF-Star-III-Chipsätze so nicht erwartet, trat jedoch auch bei späteren Untersuchungen des Empfängers in anderen Zusammenhängen auf und konnte nicht verbessert werden. Unklar bleibt, ob der Chipsatz als solches oder das darauf basierende Modul für die starken Abweichungen verantwortlich zeichnet. Mit Abweichungen im zweistelligen Meterbereich und deutlichen Empfangsproblemen in verschatteten Bereichen ist das Modul für den Ein-

satz zur Ortung von Fahrzeugen in Städten nicht geeignet. Der Venus-basierte Empfänger von Sparkfun übertrifft mit seiner hohen Präzision und der Geschwindigkeit, mit der er Veränderungen registriert, die Erwartungen. Dem realen Wegverlauf folgt er sehr genau und vor allem in schwierigen Bereichen weitaus besser als erwartet – die Abweichungen auf der Stichstraße liegen deutlich innerhalb der Toleranzen, die für eine Fahrzeugortung hingenommen werden können. Das Gerät wurde DLR-intern weiteren Tests unterzogen, welche dieses Verhalten bestätigten. Die Entscheidung über das im ersten Prototypen einzusetzende Modul fiel aufgrund der überlegenen technischen Eigenschaften zugunsten des Venus-GPS-Empfängers. Auf die mit dem späteren Prototypen erzielte Ortungsqualität wird im Kapitel 5 noch einmal näher eingegangen.

3.4. Auswahl des GSM-Moduls

Ein für den Einsatz im Rahmen dieser Arbeit geeignetes GSM-Modul hat drei wesentliche Anforderungen zu erfüllen: Es muss eine Möglichkeit zum Anschluss einer externen Antenne bieten, für alle Mobilfunkfrequenzen geeignet sein sowie den Aufbau einer paket-orientierten Datenverbindung zum Internet ermöglichen. Für Letzteres kommen daher nur Module mit *GPRS* (General Packet Radio Service) oder schnelleren Verfahren in Frage. *CSD* (Circuit Switched Data) allein reicht nicht aus, da es sich hierbei um ein leitungsvermittelteres Verfahren handelt, welches eine Relais-Stelle auf der Gegenseite benötigt. Eine Einführung in die Funktionsweise von GPRS bietet [49, S. 119 ff.]. Datenübertragungsmöglichkeiten per GPRS und *EDGE* (Enhanced Data Rates for GSM Evolution) bieten heute selbst einfache GSM-Modems, bei vielen Modellen sind außerdem weitergehende Features wie komplette Netzwerk-Protokollstacks implementiert, welche den Mikrocontroller des Ortungsmoduls massiv entlasten können. Die Unterstützung aller vier Mobilfunk-Frequenzbänder von 850, 900, 1800 und 1900 MHz (*Quad-Band*-Kompatibilität) ist hingegen selbst bei modernen Mobiltelefonen nicht immer gegeben. Für den Einsatz in tragbaren Geräten oft relevante Parameter wie der Energieverbrauch, die Baugröße und sämtliche zur Audioübertragung gehörenden Eigenschaften sind für die vorgesehene Anwendung in Fahrzeugen zunächst von untergeordneter Bedeutung, könnten aber in späteren Nutzungsszenarien mit limitierter Energieversorgung interessant werden.

Direkt für die Arduino-Plattform entwickelt waren zum Stand November 2011 nur zwei Module. Beide bieten die Möglichkeit zum Anschluss einer externen Antenne und bringen neben GPRS-Kommunikation einen vollständigen Stack für UDP, TCP und IP mit (siehe auch Anhang A, Seite A-6 ff.). Der *Arduino GSM Shield* des italienischen Anbieters *Open Electronics* besteht aus dem eigentlichen GSM-Modul sowie einer Adapterplatine, welche dessen Signalbelegung auf den Arduino-Formfaktor anpasst und dessen Spannungsversorgung sicherstellt. Diese Lösung nutzt das Funkmodul *SIM900* des chinesischen Anbieters

SIMCOM. Der alternative *Cellular Shield* der Berliner *Tinkersoup* wurde direkt für Arduinos entwickelt. Er ist ebenfalls Quad-Band-fähig und integriert einen *SM5100B-D* der chinesisch-amerikanischen *Spreadtrum Communications*.

Die Entscheidung fiel aus rein pragmatischen Gründen zugunsten des *Arduino GSM Shields*, da das Produkt von Tinkersoup zum fraglichen Zeitpunkt nicht lieferbar und rund 60% teurer war.

3.5. Zusatzmodule und Zubehör

Neben den definierten Komponenten ist die Möglichkeit zur Integration weiterer Baugruppen in das Gerät vorgesehen. Diese sollen zusätzliche Schnittstellen bieten, eine optionale Integration mit vorhandener Fahrzeugelektronik zur Gewinnung weiterer Informationen ermöglichen oder die Konfiguration und Parametrierung im Feld vereinfachen. Analog den in den vorherigen Kapiteln erläuterten Entscheidungen wurden konkrete Recherchen für folgende Zusatzkomponenten angestellt:

- SD-Karten-Anbindung für lokales Logging und Speicherung nicht in Echtzeit benötigter statistischer Daten
- Ethernet-Anbindung zur Integration mit Bussystemen einiger RBL
- Inertial Measurement Unit zur Sensordatenfusion mit dem GPS
- LCD- und Tastaturanbindung zur Interaktion mit dem Gerät ohne zusätzliche Hardware

Aus Kostengründen wurde letztendlich lediglich ein microSD-Karten-Slot mit in den Prototypen integriert. Dieser wurde in einer Demoanwendung zum GPS-Logging bereits genutzt, in der finalen Softwareversion jedoch nicht berücksichtigt (siehe Kapitel 4.3.2 sowie 4.4.1 für Details). Die vollständigen Unterlagen zur Recherche der eingesetzten sowie optionalen Komponenten finden sich in Anhang A.

Unabhängig von diesen optionalen Komponenten wurden zur Entwicklung und Nutzung des Prototypen folgende Zubehörteile beschafft:

- Steckernetzteil 12 V / 1A zur Nutzung am 230V Wechselspannungsnetz
- Kabel mit KFZ-Stecker zur Nutzung am 12V-Bordspannungsnetz in Versuchsfahrzeugen
- externer DC-DC-Wandler zur Nutzung an 24V-Bordspannungsnetzen
- Einbaugehäuse zum Schutz vor elektrostatischen Entladungen und mechanischer Einwirkung

3.6. Der erste Prototyp

3.6.1. Aufbau und Entwicklungsgeschichte

Die Entwicklung des Prototypen begann Mitte November 2011 auf Basis eines Arduino Uno. Abbildung 3.6 zeigt ihn in dem Zustand, welcher bei Abschluss dieser Arbeit erreicht war. Die im Folgenden erwähnten Komponenten sind im Bild durch Nummern gekennzeichnet.

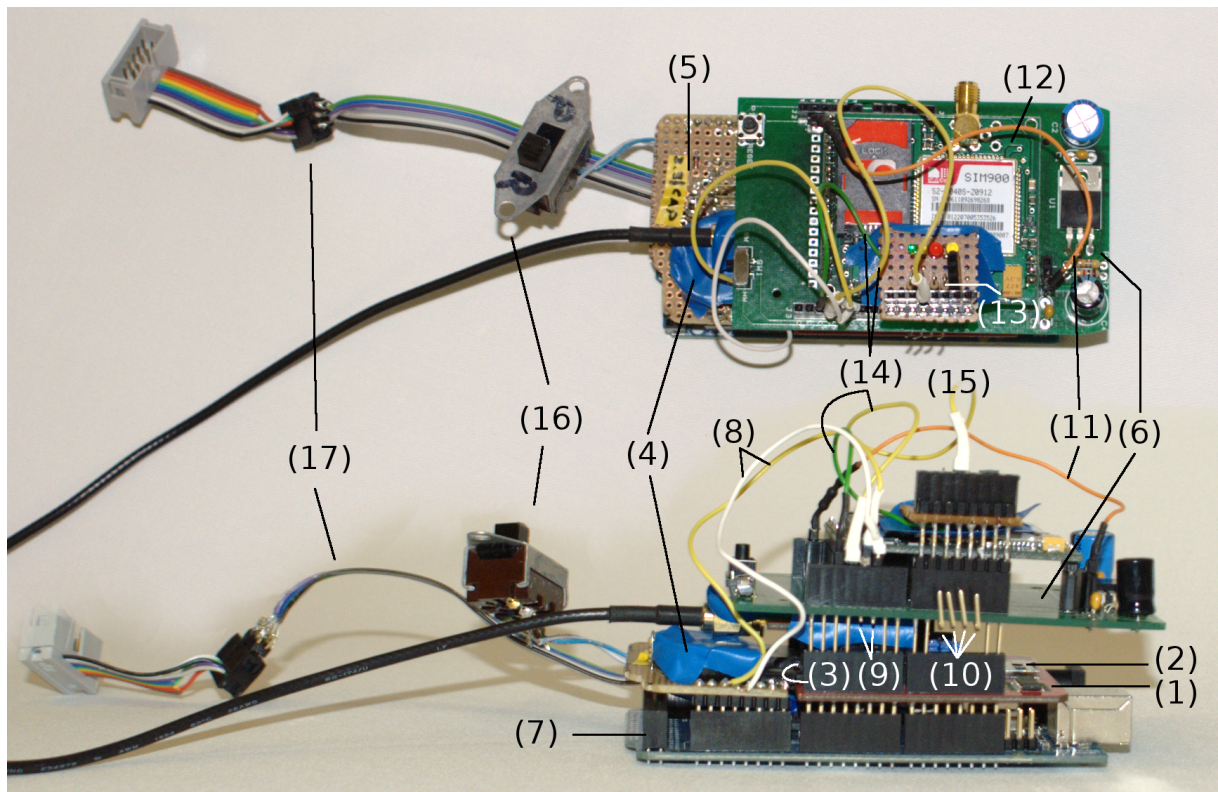


Abbildung 3.6.: Komponenten des ersten Prototypen im Endzustand

Im ersten Schritt wurde auf der freien Lochrasterfläche des beschafften microSD-Shields (1) eine Schaltung zur Pegelanpassung zwischen Mikrocontroller (CMOS 5V) und dem Venus-GPS-Modul (LVTTL 3V) aufgebaut (Schaltplan siehe Anhang C.1). Die Anbindung des GPS-Moduls an den Arduino erfolgte in diesem Szenario über eine in Software emulierte serielle Schnittstelle. Daraus resultierende Probleme mit der Geschwindigkeit erster GPS-Anwendungen und zu kleinen Schreibpuffern für den Zugriff auf die SD-Karte zeigten bereits den notwendigen Wechsel auf einen stärkeren Basiscontroller an, die Hardware arbeitete jedoch zuverlässig. Später wurde die Grundplatine mit SD-Kartenslot (2) und GPS-Modul (3) um eine Pufferspannungsversorgung (4) und eine Schnittstelle für serielle Debugger (5) erweitert.

Im zweiten Schritt wurde der GSM-Shield (6) auf dem Controllerboard Arduino Uno eingesetzt, auch hier per Anbindung über eine emulierte Schnittstelle. Auf dem Modul wurden zunächst nur GSM-Anwendungen wie SMS-Versand und Anrufe erprobt, welche eingeschränkte Anforderungen an die Anbindung zum Basiscontroller stellen. Der mangelnde RAM des kleineren Arduinos zwang dabei zu dem bereits absehbaren Wechsel auf das abgebildete größere Modell.

Der Arduino Mega 2560 (7) diente nach Abschluss der Erprobung von GSM-Anwendungen als Basiscontroller auch für GPRS-Übertragungen. Hierbei zeigten sich neben Softwareproblemen auch Unzulänglichkeiten in der Hardware v.a. des eingesetzten Arduino-GSM-Shields, welche die Entwicklung erheblich verzögerten:

- Die serielle Datenübertragung wurde auf eine der zusätzlich verfügbaren hardwareunterstützten Schnittstellen des Controllers umgestellt (8), um auftretenden Geschwindigkeitsproblemen zu begegnen. Dazu mussten die abgehenden Leitungen D4 und D5 am Modul entfernt werden (9).
- Die Pins D10 bis D13 des GSM-Shields mussten abgeklemmt werden, da sie den auf denselben Leitungen liegenden SPI-Bus massiv störten (10).
- Das Modul bietet weder eine Möglichkeit zur Erkennung des Einschaltzustandes noch einen zuverlässig flankentriggerbaren Reset-Eingang. Die zeit- und pegelgesteuerten Eingänge für beide Funktionen erfordern lange und aufwendige Einschaltprozeduren, zur Erkennung des Einschaltzustandes wurde daher eine Behelfslösung auf Basis des Netlight-Signales an Pin D6 realisiert. Dieser Pin greift sein Signal zwischen der LED D1 sowie dem Transistor T1 des GSM-Moduls ab, womit er zwischen zwei Halbleitersperrschichten liegt und im Ruhezustand kein definiertes Potential aufweisen muss. Da das Modul ohnehin mit einer niedrigeren Spannung als der Basiscontroller arbeitet, beeinträchtigt dies die korrekte Erkennung des Zustandes erheblich, zur Pegelstabilisierung wurde ein Pullup-Widerstand von 30 k Ω von Modul-Pin 19 nach Modul-Pin 1 (VCC) geschaltet (11).
- Das Modem geriet während des Betriebes in unregelmäßigen Abständen in einen undefinierten Zustand und verweigerte selbst nach Software-Resets jede weitere Kommunikation. Ursache sind zu hohe Kapazitäten an den einzelnen Leitungen der Flusskontrolle (D10 bis D13), begünstigt durch ein schlechtes Layout des Arduino-Adapterschildes unter der eigentlichen Modemplatine (12). Ladungen dieser Kapazitäten können nur schwer abgebaut werden und täuschen unter ungünstigen Umständen Steuersignale vor, welche das Modem zur Einstellung der Kommunikation veranlassen. Zur Abhilfe werden Pulldown-Widerstände mit 10 k Ω von den Leitungen gegen Masse geschaltet (13).
- Im Endstadium der Hardwareentwicklung wurde zwischen GSM-Modem und Basiscontroller eine bidirektionale Flusssteuerung unter Verwendung der zuvor entstörten

Leitungen implementiert. Dazu mussten die Signale CTS und RTS von den durch den SPI-Bus nicht verwendbaren Pins D10 und D11 umgelegt werden, für CTS wird zudem ein interruptfähiger Eingang benötigt. Gewählt wurden die noch freien Pins D2 und D3 (14), zur optischen Kontrolle wurde eine Aufsteckplatine mit Low-Current-LEDS integriert (15).

Das Arduino-Board verfügt neben dem eigentlichen Mikrocontroller über einen Zusatzprozessor, welcher die Umsetzung zwischen serieller und USB-Kommunikation übernimmt. Dieser kann auf Wunsch den Hauptcontroller bei jedem Neuaufbau einer Verbindung resetten, was über einen Lötjumper auf der Oberseite der Platine eingestellt wird. Dieser Lötjumper wurde durch einen Schiebeschalter (16) ersetzt und zusammen mit einem weiteren Kabel (ISP-Zugang, 17) aus dem Modulstapel herausgeführt, um das Laden neuer Programme unter Verwendung externer Programmieradapter (JTAG ICE mkII) zu vereinfachen.

3.6.2. Endstand

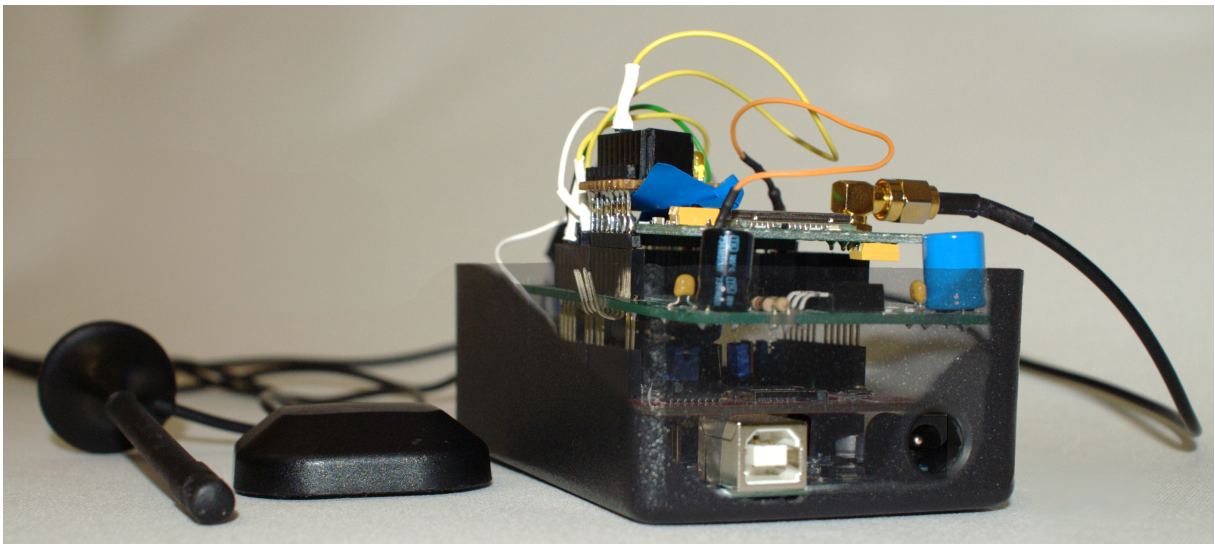


Abbildung 3.7.: Das Ortungsmodul im Endzustand. Das Schnittbild zeigt die Elektronik in der unteren Gehäuseschale, ganz links die Antenne für GSM, daneben GPS.

Abbildung 3.7 zeigt den Zustand des Prototypen zum Abschluss dieser Arbeit. Er besteht aus dem zentralen Gehäuse mit der im vorherigen Abschnitt beschriebenen elektronischen Ausstattung sowie den beiden externen Antennen. Das Gehäuse kann seit Einführung der letzten Hardwaremodifikationen nicht mehr verschlossen werden, da die Bauhöhe der obersten Platine dieses überragt. Die im Schnittbild erkennbare Elektronik im Inneren besteht von unten her aus: Arduino Mega 2560 Basiscontroller, microSD-Shield mit

GPS-Empfänger und Pegelwandler, GSM-Shield mit dem eigentlichen Modem auf einem kleineren Tochterboard und zuoberst der Anpassungsplatine für die Flusskontrolle.

Das Bild zeigt die Stärken wie auch Schwächen des modularen Prototypendesigns auf: Die einzelnen Ebenen sind ineinander gesteckt. Jede Hauptebene entspricht einer der Komponenten gemäß Abschnitt 2.5.2, lediglich die Energieversorgung ist aufgrund der verschiedenartigen Anforderungen der Module dezentralisiert. Trennung, Wartung, Modifikation und Austausch einzelner Komponenten sind sehr einfach, was insbesondere die Fehlersuche während der Hardwareentwicklung erleichtert. Dafür erfordert diese Bauweise viel Platz, das Gehäusevolumen wird sehr ineffizient genutzt. Die Module waren von ihren Herstellern aus nie für einen gemeinsamen Gehäuseeinbau vorgesehen, weswegen die relevanten Anschlüsse weder auf einer gemeinsamen Seite noch in identischer Bautiefe liegen, was die Fertigung selbst einfacher Frontplatten stark erschwert. Bauteile wie Pegelwandler-ICs sind oft mehrfach vorhanden, werden aber jeweils nur partiell genutzt, da jede Komponente ihre eigene Basiselektronik mitbringt. Die Modifikationen und Fehlerbehebungen an der Hardware insbesondere der Mobilfunkkomponente erfordern Neuverdrahtungen, welche als freie Leitungen über den Modulen verlaufen. Die Gefahr der ungewollten Trennung oder Beschädigung dieser Leitungen ist hoch, die Kombination vieler Steckverbindungen birgt ein hohes Risiko des Auftretens von Störungen durch Wackelkontakte oder brechende Lötstellen. Wichtige Punkte wie der Reset-Lötjumper des USB-seriell-Wandlers, der ISP-Anschluss des Basiscontrollers oder dessen Resetknopf sind nicht mehr einfach zugänglich.

Das modulare Design ist somit zur Evaluierung und Entwicklung der ersten Stufen neuartiger Systeme geeignet, sollte für den Praxiseinsatz jedoch durch optimierte und höher integrierte Lösungen ersetzt werden. Mit dieser nächsten Evolutionsstufe befasst sich der Abschnitt 7.1.1.

4. Software

4.1. Entwicklungskonzept

Die Entwicklung eingebetteter Systeme verläuft nicht grundlegend anders als Softwareprojekte in definierten Umgebungen (Rechner mit abstrahierendem Betriebssystem). Durch die eingeschränkten Möglichkeiten der eher leistungsschwachen Prozessorkerne und die enge Verzahnung mit der sie umgebenden Hardware haben sich jedoch bestimmte Ansätze und Vorgehensmodelle immer wieder bewährt.

Zu unterscheiden ist, ob es in einem Projekt um die reine Entwicklung neuer Software für fertig entwickelte, getestete und integrierte Hardware oder um den Entwurf gänzlich neuer Systeme geht. Insbesondere die parallele Entwicklung von Hard- und Software birgt zahlreiche Unsicherheiten und Unwägbarkeiten, die Einfluss auf den Entwicklungsprozess haben.

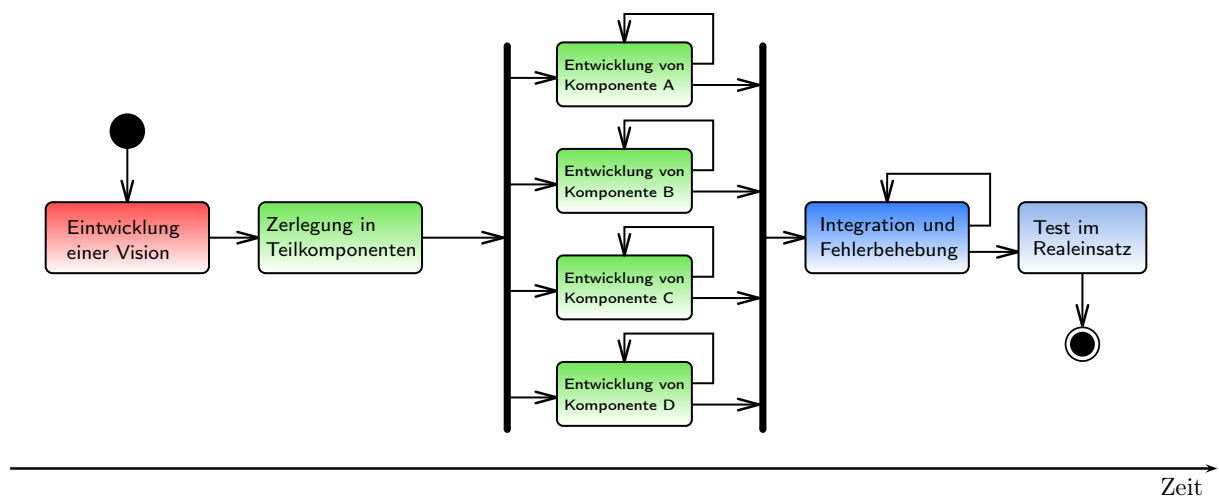


Abbildung 4.1.: Entwicklungsprozess eingebetteter Systeme als Aktivitätsdiagramm

Bei der Entwicklung *from Scratch* wird zunächst eine Vision definiert, eine Idee, was das fertige System leisten und wofür es eingesetzt werden soll. Dies ermöglicht die Aufteilung der Gesamtlasten in mehrere zunächst unabhängige Teilaufgaben, welche separat bearbeitet werden können. Erst nachdem jede Komponente für sich funktionsfähig ist, wird das

System integriert und zu seiner endgültigen Funktion geführt. Die Entwicklung der Teilkomponenten umfasst dabei eine Abschätzung der erforderlichen Leistungsfähigkeit der genutzten Controller (etwa für aufwändige Filtersysteme), die Auswahl geeigneter Hardware und die Implementierung der jeweiligen Lösung. Sollen Ressourcen (etwa Energie, Rechenleistung oder Speicher) zwischen mehreren Komponenten geteilt werden, ist eine Absprache der Beteiligten erforderlich. Die Integration stellt aufgrund der überschaubaren Größe und Komplexität der Systeme eine geringere Herausforderung als in großen Softwareprojekten dar. Damit liegt der Schwerpunkt des in Abbildung 4.1 dargestellten Entwicklungsprozesses auf den grün hervorgehobenen Aufgaben.

Die Entwicklung von Software auf *fertiger* (vorhandener, bekannter und getesteter) Hardware hingegen verfolgt zumeist das Ziel einer erheblichen Leistungssteigerung und besseren bzw. geänderten Ausnutzung der gegebenen Ressourcen. Hierbei kommt es insbesondere auf eine genaue Bestimmung der Kennwerte an, welche vom *integrierten Gesamtsystem* nicht überschritten werden dürfen (Laufzeit, Speicherverbrauch, Energieaufnahme). Bereits existente Teilkomponenten werden oft wiederverwendet und gemäß der neuen Anforderungen optimiert, sind in ihrer Funktionalität aber schon gegeben oder konnten aufgrund der Überschreitung der Maximalwerte diese nicht entfalten. Das Hauptaugenmerk liegt daher viel stärker auf der Integration der Komponenten, deren Zusammenspiel unter allen geforderten Bedingungen (Ressourcenverbrauch im Fehlerfalle) gesichert werden muss. Das Hauptaugenmerk des Entwicklungsprozesses fällt damit in den in Abbildung 4.1 blau markierten Bereich.

Im Rahmen dieser Arbeit ging es um die komplette Neuentwicklung eines Gerätes. Die Vision war mit der Aufgabenstellung weitestgehend gegeben, die einzusetzenden Hardwarekomponenten aber waren noch nicht bestimmt und dementsprechend vorher nie miteinander integriert worden. Software stand nur in Form von Beispielprogrammen und Bibliotheken für einige Basisfunktionen zur Verfügung. Trotz des bekannten Zieles der Entwicklung gestaltet sich eine detaillierte Vorab-Spezifikation für solche Entwicklungsvorhaben schwierig, da die enge Verzahnung von Hard- und Software Unsicherheiten und Unwägbarkeiten mit sich bringt, welche in der klassischen Softwareentwicklung (nur *innerhalb* eines Rechners) nicht existieren. Damit stellt sich auch die Frage nach einem geeigneten Vorgehensmodell zur Softwareentwicklung unter etwas anderen Gesichtspunkten.

Der geringe Codeumfang der Einzelkomponenten eingebetteter Systeme fördert ständige Builds zur Sicherstellung der Lauffähigkeit des erzeugten Codes – Kompilat im Bereich einiger zehn bis weniger hundert Kilobyte ist auch an normalen Entwicklerrechnern schnell erstellt und erfordert keine aufwändige Build-Infrastruktur. Häufige Tests auch auf der realen, ggf. selbst noch im Entwicklungsstadium befindlichen Hardware sind eher die Regel als die Ausnahme. Fehler und unerwartete Verhaltensweisen der Hardware, welche Pläne zur Anbindung ganzer Komponenten blockieren können, sind durchaus zu erwarten.

Dementsprechend drängt sich ein iteratives Vorgehen, welches Änderungen und Überarbeitungen der Spezifikation auch in einem späten Entwicklungsstadium noch zulässt, geradezu auf. Der durch das zuvor beschriebene Modell eines Entwicklungsprozesses bereits sehr naheliegende Bottom-Up-Ansatz und die Teamstärke von nur einem Entwickler sowie wenigen Stakeholdern ermöglichten im konkreten Fall ein sehr agiles Vorgehen bis hin zum Extreme Programming. Die von Beginn an festgelegte Laufzeit von drei Monaten zur Fertigstellung nicht nur des ersten Prototypen, sondern auch dieser Dokumentation sowie die bestehenden Pläne zur anschließenden Weiterentwicklung des Gerätes stehen dem XP-typischen Verzicht auf weite Teile der Dokumentation jedoch entgegen, so dass im Rahmen einer auf schnellen Fortschritt ausgerichteten Hard- und Softwareentwicklung auch regelmäßige, umfassende Refactorings durchgeführt und zu den jeweils bearbeiteten Teilaufgaben wichtige Informationen zusammengefasst und notiert wurden.

4.2. Gegebene Softwarekomponenten

4.2.1. Die Arduino-Umgebung

Das Arduino-Projekt, welches bereits in Abschnitt 3.2 vorgestellt wurde, stellt neben Hard- und Software der Mikrocontroller auch eine Entwicklungsumgebung bereit. Diese bildet ein sehr monolithisches Werkzeug, welches seine Interna und insbesondere den Build-Prozess des Quellcodes vor den Benutzern zu verstecken versucht. Für die Hauptzielgruppe des Projektes durchaus angemessen, bereitet dies für fortgeschrittene Anwendungen wie die Entwicklung des Ortungsmoduls erhebliche Schwierigkeiten.

Die Arduino-IDE unterstützt nur eine primitive Syntaxhervorhebung und keinerlei Features wie Autovervollständigung, Refactoring, Auto-Formatierung oder Codeverfolgung. Anstatt normalem C- und C++-Code werden lediglich Methodenrümpfe geschrieben, für die ein in der IDE fest integrierter Präprozessor zunächst Prototypen und Header generiert, bevor die eigentliche Kompilierung beginnt. Dies macht die Nutzung bestimmter Peripherie und fortgeschrittenes Debugging der AVR-Mikrocontroller unmöglich, da dafür sogenannte Init-Sequenzen noch vor dem eigentlichen Boot und lange vor dem Einsprung in die C-übliche `main()` geladen werden müssen. Ohne Zugriff auf die Parametrierung der am Build-Prozess beteiligten Programme sind Bibliotheken, welche Änderungen dieser Einstellungen oder das Linken gegen eigene Objekte erfordern, schlicht nicht nutzbar (etwa `printf()` für Gleitkommazahlen). Daher wurde im Verlauf der Arbeiten ein Wechsel auf geeignetere Software erforderlich. Um die Komponenten des Arduino-Betriebssystems dennoch weiterhin nutzen zu können, mussten aufwändige Anpassungen der sonst für AVR-Controller üblichen Makefiles vorgenommen werden. Das unter Nutzung der Vorarbeiten Dritter entstandene Script ist im Hauptverzeichnis der geschaffenen Software im elektronischen Anhang zu finden.

Das dank dieser Anpassungen weiterhin nutzbare Arduino-Betriebssystem bietet einen Abstraktionslayer oberhalb der Peripheriemodule des Mikrocontrollers. Dazu werden einzelne Klassen für jedes genutzte Modul im globalen Speicherraum des Controllers präinstanziiert, welche standardisierte oder zumindest konzeptuell vereinheitlichte Möglichkeiten zum Zugriff auf verschiedene Bussysteme, zur Formatierung von ASCII-codierten Zeichenketten und zum Setzen häufig genutzter Konfigurationsregister bieten. Dies ermöglicht auch das Erstellen zusätzlicher Bibliotheken, welche aus Sicht des Benutzenden nach demselben Schema wie die mitgelieferten Kernkomponenten angesprochen werden können, so sie in den entsprechenden Linker-Aufrufen gelistet werden. Die nachfolgend beschriebenen Programmteile machen davon ebenso Gebrauch wie im Rahmen dieser Arbeit selbst erstellte Software.

4.2.2. GPS-Bibliothek

GPS-Module werden vor allem in eingebetteten Systemen beinahe immer über einfache serielle Schnittstellen angebunden. Neben den binären, proprietären Protokollen der Hersteller bieten sie die Ausgabe der verfügbaren Informationen über das sogenannte NMEA-Protokoll an. Dieses standardisiert den Aufbau der Nachrichten und die zu sendenden Datenformate. Die Version NMEA-0183 von 1983 war auf die Leistungsfähigkeit damaliger Prozessoren angepasst, was sie heute auch mit einfachen Mikrocontrollern lesbar macht. Sie wird daher (im Gegensatz zu neueren Versionen desselben Protokolls) von beinahe allen Empfängern unterstützt. Die Bibliothek *TinyGPS* von Mikal Hart beherrscht das Parsen dieses Protokolls sowie die Prüfung der Checksummen jedes Datensatzes [40]. Sie steht unter der GNU LGPL 2.1+ und kann unabhängig vom konkreten Empfänger GPS-Positionsdaten in einer für den Controller verwertbaren Form bereitstellen – im Rahmen dieser Arbeit etwa als einfache Gleitkommazahlen, welche via GPRS an den AIM-Webservice übertragen werden können.

4.2.3. GSM/GPRS-Bibliothek

Open Electronics bietet für den in dieser Arbeit eingesetzten Arduino GSM Shield eine Library an, welche die Funktionen des verwendeten SIMCOM-Modules unterstützen soll [41]. Dessen Ansteuerung funktioniert über den de-facto-Standard der Modemkommunikation, den AT-Befehlssatz. Die Kommunikation zwischen Modem und Hauptcontroller erfolgt bidirektional und unterliegt insbesondere sehr präzisen Timinganforderungen, weswegen eine Abstraktion derselben zwingend notwendig ist. Die zu Beginn dieser Arbeit vorliegende Version unterstützte lediglich wenige Funktionen für Telefonie und SMS-Verarbeitung, es wurden seit November 2011 jedoch mehr als 10 neue Versionen

veröffentlicht. Die Mehrzahl derselben befindet sich unter [42]. Obwohl der Funktionsumfang dieser Bibliothek von den Entwicklern beständig erweitert wird, bleibt deren Zeitplan deutlich hinter den Erwartungen zurück. Features waren und sind nicht zu den benötigten Zeitpunkten verfügbar und Bugs wurden und werden nur mit hoher Verzögerung behoben. In der aktuellen Firmware des Ortungsmoduls kommt diese Bibliothek daher überhaupt nicht mehr zum Einsatz, sie spielte während der Erprobung des GSM-Shields und für die Entwicklung einer eigenen Alternative jedoch eine bedeutende Rolle (siehe auch Abschnitt 4.6.2).

4.3. Entwicklung erster Testprogramme

4.3.1. GPS-Benutzung

GPS-Module stellen wie in Kapitel 4.2.2 beschrieben in der normalen Anwendung reine Sender dar. Daten *zum* Modul werden nur für einmalige Konfigurationszwecke gesandt, was in der Regel die Benutzung einer proprietären Software des jeweiligen Herstellers erfordert, im Einsatz aber nie vorkommt. Daher gestaltet sich die Anwendungsentwicklung relativ einfach.

Erste Testprogramme nutzten auf dem Arduino Uno noch eine in Software implementierte serielle Schnittstelle, mit der Verwendung des Mega 2560 wurde diese durch eine hardwareunterstützte Lösung ersetzt. Neben der bereits erwähnten TinyGPS-Bibliothek zum Parsen der NMEA-Datensätze wurde auch eine einfache Relais-Software geschrieben, welche die Daten vom GPS-Modul an einen an den Mikrocontroller angeschlossenen Rechner weiterleitet und umgekehrt. Dieses Programm dient nach wie vor zum Direktzugriff auf die NMEA-Datensätze, da diese nur so ohne einen Ausbau des Moduls aus dem Prototypen zugänglich sind. Ein weiteres Experiment demonstrierte die Möglichkeit zum Mitschreiben der Datensätze auf SD-Karte (siehe nächster Abschnitt). All diese Programme sind dank des einheitlichen Aufbaus der gesendeten Nachrichten sowohl mit dem Venus-GPS als auch dem MOD-GPS verwendbar.

4.3.2. SD-Logging

Parallel zur Erprobung der GPS-Empfänger wurde auf dem Arduino Uno die Anbindung einer microSD-Karte demonstriert. Dazu kommt eine der betriebssystemeigenen Standardbibliotheken zum Einsatz, welche den Hardware-SPI-Bus des Mikrocontrollers ansteuert. Dieser wurde ursprünglich für die älteren und viel langsameren MMC-Karten genutzt, zu denen SD-Karten und ihre Nachfolger bis heute abwärtskompatibel sind. Die Software

konnte durch Verwendung eigener Puffer Routinen so weit optimiert werden, dass trotz der niedrigen Zugriffsgeschwindigkeiten auf die Karte Datensätze bis 9600 Baud und 1 Hz Updaterate aufgezeichnet werden können. Gelegentlich auftretende Datenfehler könnten durch eine Erhöhung der Bustaktfrequenz und größere Puffer auf dem Mega 2560 vermieden werden, das Programm wurde mangels Notwendigkeit bisher jedoch nicht portiert. Eine vollständige Unterstützung der Karten für Debugzwecke und zur Aufzeichnung statistischer Daten soll im Zuge der Weiterentwicklung des Ortungsmoduls integriert werden.

4.3.3. GSM- und GPRS-funktionen

4.3.3.1. GSM-Telefoniefunktionen

Die ersten Versionen der zum Ardiono GSM Shield gehörenden GSM-Bibliothek unterstützten lediglich einige Modemfunktionen aus dem Bereich der klassischen Mobiltelefone. Diese stellen geringere Anforderungen an die serielle Verbindung zwischen Controller und Modem – einem der wesentlichen Problempunkte bis zum Abschluss dieser Arbeit. Die mit diesen Versionen geschriebenen Versuchsanwendungen testen den Zugriff auf Telefonbuchfunktionen und SMS-Speicher der SIM-Karte, den Empfang und Versand von Textnachrichten sowie Anrufe auf andere Telefone (ohne Audio-Ein-/Ausgabe). Um die Beschränkungen der Bibliothek zu umgehen und den direkten Internetzugriff vorzubereiten wurde der mitgelieferte Quelltext von Open Electronics erstmalig massiv erweitert. Die größten hinzugefügten Routinen gestatten das freie Senden beliebiger AT-Kommandos und die Untersuchung des Antwortverhaltens des Modems. Die Integration dieser Teilprogramme in eine komfortabler nutzbare, per Textmenü geführte Software bereitete erhebliche Schwierigkeiten, da der RAM des Mikrocontrollers zu klein war und Heap und Stack unkontrolliert ineinander liefen.

4.3.3.2. GPRS-Datenkommunikation

Mit dem Wechsel auf den Mega 2560 konnten die zuvor mangels Speicher nicht möglichen Telefonieanwendungen und Versuche mit neu zugänglichen Features realisiert werden, womit die meisten Testprogramme der Entwicklung der offiziellen Library deutlich voraus waren. Anfragen zur Zusammenarbeit blieben seitens der GSM-Shield-Entwickler unbeantwortet.

Die Unterstützung zum Aufbau von Datenverbindungen wurde im Rahmen dieser Tests analog zu klassischen (leitungsgebundenen) Modems so weit ausgebaut, dass Verbindungen auf Ebene 2 gemäß OSI-Referenzmodell aufgebaut werden können. Die Gegenstellen im Mobilfunknetz leiten jedoch nur Datenverkehr der Ebene 3 und höher weiter, weswegen

über diese Verbindung noch keine Nutzdaten ausgetauscht werden können. Fast zeitgleich mit der Fertigstellung dieser Testprogramme wurde eine neue Version der GSM-Bibliothek veröffentlicht, welche die im verwendeten SIM900-Modul vorhandenen Möglichkeiten zum direkten Aufbau von IP-Verbindungen auf Layer 3 zugänglich macht – eine nicht im AT-Standard vorgesehene, aber übliche Erweiterung, für die jeder Hersteller selbst definierte, an den AT-Standard angelehnte Kommandos nutzt. Damit entfällt die Notwendigkeit zur Implementierung eines TCP/IP-Stacks auf dem Hauptcontroller, was diesen wesentlich entlastet. Nachfolgende Tests dienten primär der Beseitigung von Hardwareproblemen mit dem Shield.

4.4. Firmware des Gerätes: Anforderungen und geplanter Funktionsumfang

4.4.1. Ortung, Logging und Datenaufbereitung

Das Gerät soll seine aktuelle Position kontinuierlich an einen Webservice übermitteln können. Dazu muss es in der Lage sein, diese zu erfassen, was nicht nur eine rein sensorische Herausforderung darstellt. Messwerte sind unabhängig von der Art ihrer Gewinnung stets fehlerbehaftet, es gilt daher, diese mittels geeigneter Verfahren zu überprüfen und im Einzelfalle zu entscheiden, ob eine Information eine hinreichend genaue Lokalisierung für den angestrebten Einsatzzweck zulässt. Es sollen nur solche Informationen an den Webserver übertragen werden, die für die unmittelbare Anwendung erforderlich sind. Je nach eingesetztem Ortungsverfahren liegen jedoch noch weitere, über den unmittelbaren Zweck der Lokalisierung hinausgehende Informationen vor, deren Auswertung im Echtzeiteinsatz nicht erforderlich ist, zur Gewinnung wissenschaftlicher Daten oder zur Bewertung der Leistungsfähigkeit des Gesamtsystems aber dennoch interessant sein kann. Diese sollen daher – vorzugsweise lokal – vorgehalten werden, um bedarfsabhängig ausgelesen werden zu können.

4.4.2. Konfiguration

Für den parallelen Einsatz mehrerer Geräte muss eine Reihe von Parametern konfigurierbar sein. Dies betrifft etwa Zugangsdaten für Kommunikationsgegenstellen (Webservices), die Auswahl der zu erfassenden Daten (z.B. Position, Geschwindigkeit, Anzahl Haltestellen, ...) sowie pro Gerät eindeutige Identifikationsnummern, welche nicht nur das verwendete Fahrzeug, sondern auch dessen Aufgabe (z.B. Busfahrt auf einer bestimmten Linie) kenntlich machen. All diese Parameter müssen auf eine Weise vorgehalten werden,

welche Änderungen im eingebauten Zustand und ohne aufwändige externe Hardware wie Programmieradapter zulässt. Notwendige regelmäßige Eingaben dürfen den normalen Betriebsablauf in der Einsatzumgebung nicht signifikant stören, Fehlbedienungen nicht zum Verlust der Funktionsfähigkeit des Gerätes führen. Dabei sind Rückmeldungen an das Bedienpersonal erforderlich, die den Erfolg ausgeführter Aktionen signalisieren.

4.4.3. Kommunikation und Datenübertragung

Die Übertragung ausgewählter Daten zu einem entfernt laufenden Webservice ist zentraler Bestandteil des Einsatzzweckes des Ortungsmoduls. Die Kommunikation ist die aufwändigste aller spezifizierten Teilkomponenten, darf jedoch die Erfüllung der übrigen Aufgaben nicht behindern. Gleichzeitig stellen Funknetzwerke keine zuverlässigen Medien dar, deren Verfügbarkeit zu jedem Zeitpunkt sichergestellt ist. In öffentlichen Netzen agierende Komponenten bieten je nach Ausführung eine erhebliche Angriffsfläche für unerwünschte Aktionen Dritter, welche es auszuschließen gilt. Die Behandlung auftretender Fehler sowie die Sicherstellung der Integrität und Vertrauenswürdigkeit übermittelter Daten stellen wichtige Planungsgrößen dar, deren Umsetzung zumindest mit vorgesehen werden muss.

4.5. Die AIM-Webservices als Datenempfänger

Bei den Webservices handelt es sich um einen Teil des AIM-Projektes, welcher als Schnittstelle zwischen dezentral eingesetzten Komponenten und den stationären Hauptsystemen fungiert. Sie befinden sich überwiegend noch im Spezifikations- und Planungsstadium und sind nicht für die reale Nutzung verfügbar, sollen jedoch unter anderem als spätere Gegenstelle der Ortungsmodule agieren und von diesen gesendete Informationen in die zentralen Datenbanken aufnehmen.

Zur Ermöglichung erster Tests wurde daher von DLR-Mitarbeitern eine einfache HTTP-Schnittstelle geschaffen, welche diese Lücke ausfüllt. Ein Webserver mit darauf laufenden PHP-Anwendungen und einer dahinter stehenden MySQL-Datenbank ermöglicht das Stellen einer Reihe verschiedener GET-Requests, womit nicht nur neue Daten in Form von GET-Parametern in diese Datenbank eingepflegt, sondern über entsprechende Schlüssel auch wieder ausgelesen werden können. Ein System aus Benutzernamen und gehashten Passwörtern bietet dabei einen Basis-Schutz vor unbefugten Zugriffen. Die HTTP-Schnittstelle wurde parallel zu den Arbeiten am Ortungsmodul optimiert, um das je Übertragung notwendige Datenvolumen zu minimieren und die Zugriffsgeschwindigkeit erhöhen zu können.

Dieses System soll jedoch nicht für die geplante Erfassung realer Daten einer Vielzahl von Fahrzeugen dienen und wird daher ersetzt werden. Im Zuge dieser Ersetzung ist

auch eine grundlegende Neugestaltung der Schnittstellen zu erwarten, um diese besser an die Bedürfnisse der Ortungsmodule anzupassen und die Übertragung gewonnener Daten weiter zu optimieren. Daher sind derzeit noch keine schnittstellenabhängigen Bestandteile der Kommunikationsimplementierung zu realisieren.

4.6. Entwicklung der Gerätefirmware

4.6.1. Konzeption

4.6.1.1. Überblick

Die Software stellt in eingebetteten Systemen eine Teilkomponente dar, der mitunter ein hoher Grad an Vermittlungsfunktionen zukommt. Beschränkt auf die Ausführung in je einem Mikrocontroller, welcher mit weiteren Baugruppen in seiner Umgebung interagieren muss, unterscheiden sich die Möglichkeiten zur Realisierung von Objekt- und Kontrollflüssen gerade zwischen verschiedenen *intelligenten* Komponenten erheblich von modernen Betriebssystemen, auf denen Multithreading und virtuelle Speicherverwaltungen die Nebenläufigkeit verschiedener Prozesse gestatten.

Das Ortungsmodul verfügt über insgesamt drei solche intelligenten Komponenten: Den GPS-Empfänger, den Hauptcontroller sowie das GSM-Modem (vgl. auch das Komponentendiagramm 2.5 des Moduls in Kapitel 2.5.2). Das Zusammenwirken dieser Komponenten ist im Interaktionsübersichtsdiagramm Abb. 4.2 dargestellt, wobei die Grundaufgaben der Gerätefirmware des Hauptcontrollers noch einmal im Abschnitt 4.6.1.3 und dort insbesondere im Aktivitätsdiagramm Abb. 4.4 präzisiert werden.

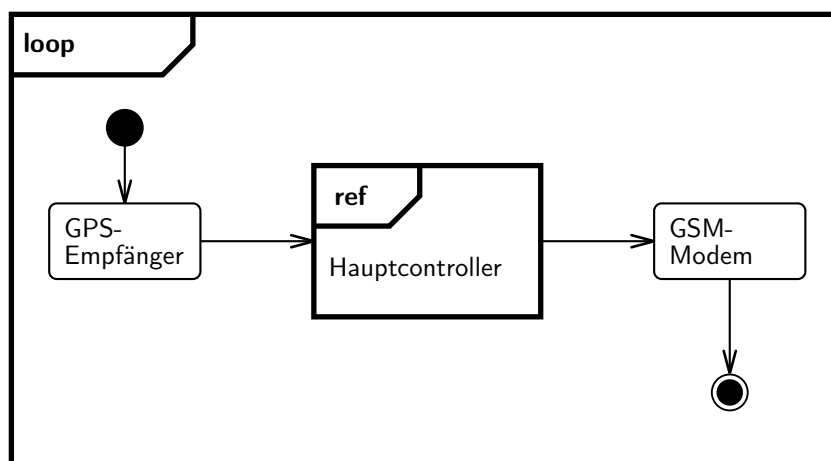


Abbildung 4.2.: Interaktionsübersichtsdiagramm der Komponenten des Ortungsmoduls

4.6.1.2. Kommunikation zwischen beteiligten Komponenten

Mit der Verwendung mehrerer softwaregesteuerter Komponenten in einem Gesamtsystem ist auch ein Datenaustausch zwischen diesen erforderlich. Für Ortungssysteme nach dem im Rahmen dieser Arbeit definierten Aufbau bedeutet dies das Auftreten zweier Übertragungstrecken: Eine zwischen GPS-Empfänger und Hauptcontroller sowie eine zwischen Hauptcontroller und GSM-Modem. Der Hauptcontroller synchronisiert dabei die Abläufe zwischen allen Teilsystemen und übernimmt weitere Aufgaben insbesondere in der Aufbereitung und Verarbeitung von Daten.

Auf einer dieser Übertragungstrecken sendet ein GPS-Empfänger zyklisch NMEA-Datensätze. Er wartet dafür nicht auf Anfragen etwaiger Kommunikationspartner, sondern sendet in regelmäßigen Intervallen. Daher liegt eine der Aufgaben des Hauptcontrollers darin, den Beginn neuer Datensätze zu erkennen und diese zu verarbeiten, so er Informationen benötigt, oder diese ansonsten zu verwerfen.

Die Kommunikation auf der anderen Übertragungstrecke – zwischen Controller und Modem – erfolgt bidirektional und ist deutlich komplexer, da letzteres indirekt die Kontaktaufnahme zu einer weiteren Komponente – dem AIM-Webservice – ermöglicht. Damit erhöht sich der zu betreibende Aufwand weiter. Paket- und verbindungsorientierte Übertragungen wie diese erfordern unabhängig von der gewählten Hardware immer eine Abfolge von mindestens drei Schritten: Verbindungsaufbau, Senden der Daten, Verbindungsabbau. Kommt wie hier noch eine Zwischenkomponente als Vermittlerin hinzu (das Modem), fallen gegebenenfalls zusätzliche Schritte zur Steuerung derselben an. Dementsprechend hoch ist der Kommunikationsaufwand für diese Strecke und entsprechend groß der Anteil der dort stattfindenden Kommunikation am insgesamt auftretenden Transfer-volumen. Diese wesentlichen Vorkommnisse auf beiden Übertragungstrecken und ihre zeitliche Abfolge werden schematisch auch noch einmal im Sequenzdiagramm Abb. 4.3 dargestellt.

4.6.1.3. Datenverarbeitung im Mikrocontroller

Die Software innerhalb des Hauptcontrollers ist in diesem Szenario die einzig beeinflussbare unter den gegebenen Komponenten. Gemäß den Erläuterungen in den vorherigen Abschnitten kommen ihr Aufgaben der Datenprüfung und -verarbeitung sowie der Kommunikation zu. Implementierungsunabhängig bedeutet dies, dass bestimmte Prozesse wiederkehrend durchlaufen und Entscheidungen zum Umgang mit der jeweils aktuellen Situation getroffen werden müssen – der Verlaufspfad des *Normalbetriebes* folgt immer gleichen Grundsätzen und ist im Aktivitätsdiagramm Abb. 4.4 dargestellt. Darin enthalten sind insbesondere zwei Stufen: Der Empfang der Daten vom GPS mitsamt der zugehörigen Aufbereitung sowie das Senden der Daten zum Webservice.

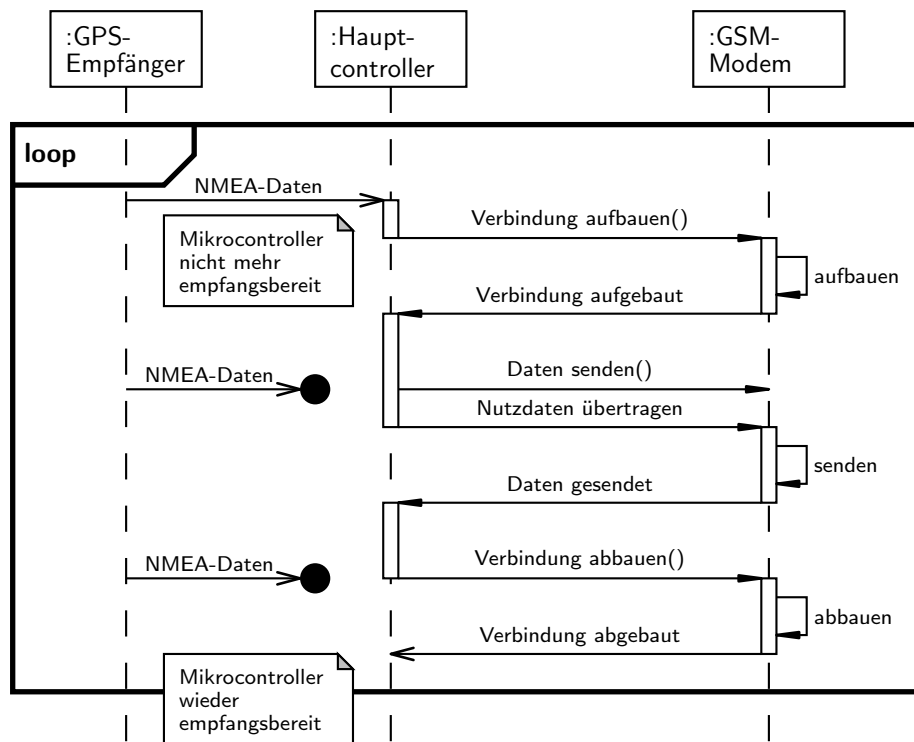


Abbildung 4.3.: Sequenzdiagramm zum Kommunikationsablauf innerhalb des Ortungsmoduls

Die möglichen Ausnahme- und Fehlersituationen hingegen hängen stark von den Verhaltens- und Signalisierungsmöglichkeiten der jeweiligen Hardware ab, sie sind daher in der Abbildung nicht mit berücksichtigt. Das Diagramm stellt auch eine Präzisierung der Komponente Hauptcontroller aus Abb. 4.2 dar.

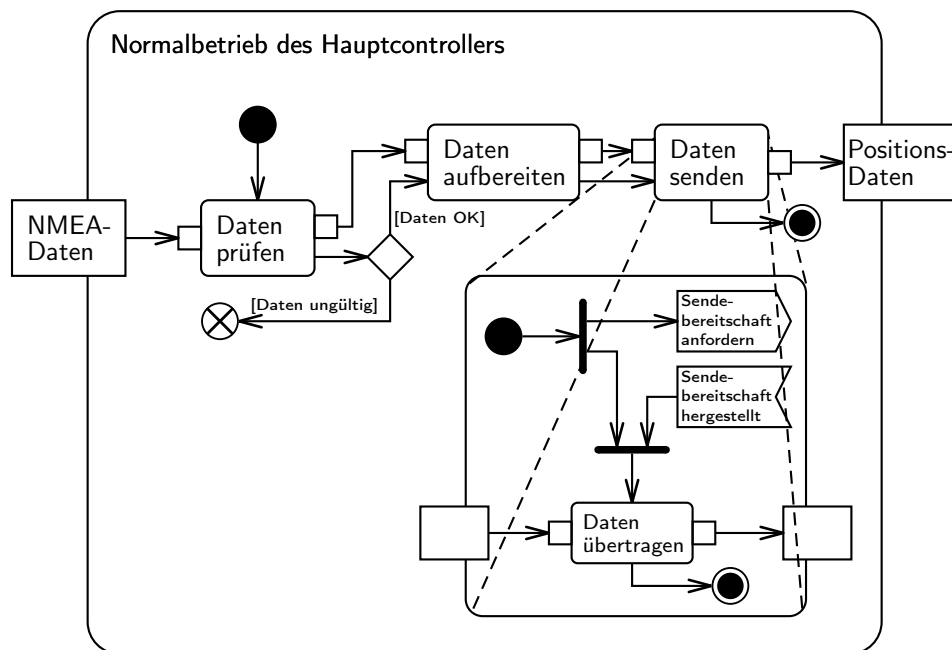


Abbildung 4.4.: Aktivitätsdiagramm zum normalen Betriebsablauf im Hauptcontroller

4.6.1.4. Abstraktion der Hardware durch Klassen

Der Hauptcontroller bindet als zentrale Komponente des Ortungsmoduls andere Baugruppen an. Dies geschieht über die Kommunikationsstrecken, deren prinzipielle Aufgaben in den vorherigen Abschnitten beschrieben wurden. Für eine Implementierung muss diese abstrakte Definition auf reale Befehle abgebildet werden, logische Aufgaben werden durch Folgen konkreter Kommandos ersetzt. Für die Wartbarkeit und Flexibilität der Software ist es dabei unerlässlich, diese in einzelne Module zu gliedern und sowohl nach der jeweils angebotenen Komponente als auch nach der Abstraktionsebene verschiedener Quelltextbestandteile zu trennen. Das Ziel ist es, beliebige Teile des Systems austauschen und verändern zu können, ohne nicht direkt von der jeweiligen Hardware abhängige Softwarebestandteile modifizieren zu müssen. Abbildung 4.5 nutzt die Beschreibung mittels Komponenten und Objekten, um diese Trennung schematisch aufzuzeigen. Mit der realen Implementierung beschäftigt sich Kapitel 4.6.2.

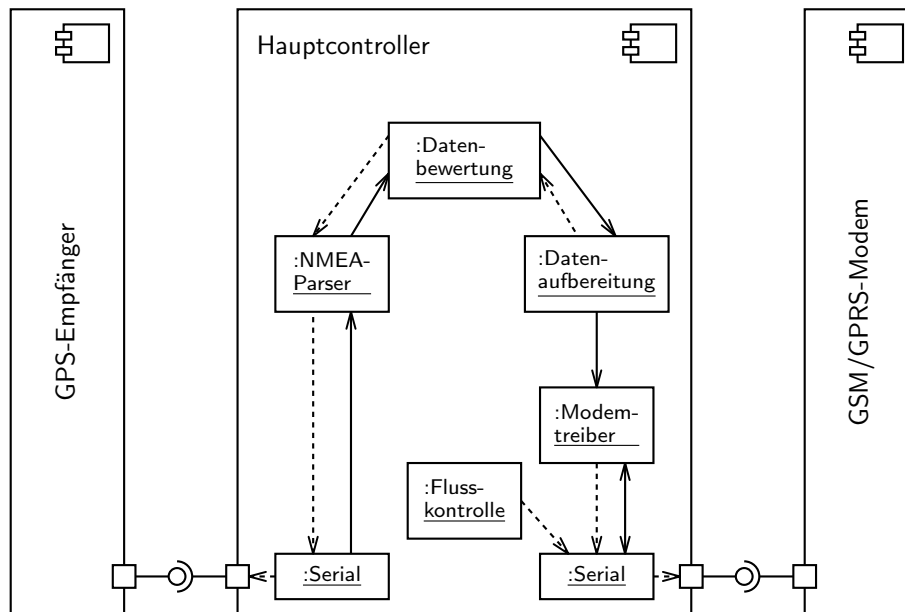


Abbildung 4.5.: Objekt- und Komponentendiagramm zur Modularisierung und Abstraktion der Software im Hauptcontroller

4.6.2. Implementierung

4.6.2.1. Struktur

Die Implementierung der Firmware des Ortungsmoduls richtet sich nach dem in Kapitel 4.6.1 beschriebenen Modell. Das Programm im Hauptcontroller ist als Endlosschleife realisiert, welche fortwährend Daten vom GPS-Empfänger annimmt und (nach mehreren Zwischenschritten) über das GSM/GPRS-Modem wieder abgibt (vgl. Abb. 4.2). Die Schrittfolge eines einzelnen Schleifendurchlaufes folgt dabei dem Konzept nach Abb. 4.4.

Der Datenfluss eines jeden Durchlaufes gliedert sich in äußere und innere Kommunikation. Äußere Kommunikation bezeichnet den Datentransfer über physische Schnittstellen von und zu den weiteren Komponenten, dieser findet zu Beginn und Ende eines Durchlaufes statt. Der Zugriff auf die externen Komponenten wird dabei über mehrere Ebenen abstrahiert, so dass der Kern der Software die ursprüngliche Quelle seiner Daten weder bestimmen kann noch muss. Innere Kommunikation meint den Datenaustausch zwischen den verschiedenen Softwaremodulen der Firmware. Diese findet ausschließlich über den internen flüchtigen Speicher des Hauptcontrollers statt und ist von außen nicht zugänglich.

Diese Sachverhalte versucht Abbildung 4.6 darzustellen: Dort ist die wesentliche Kommunikationsstruktur der Software abgebildet. Die Programmschleife (`loop()`) befindet sich im `main`-Objekt (grün), sie steuert die Abfolge sämtlicher Vorgänge. Der Kontrollfluss folgt den Assoziationen (durchgezogene Pfeile), so dass ein Schleifendurchlauf in der

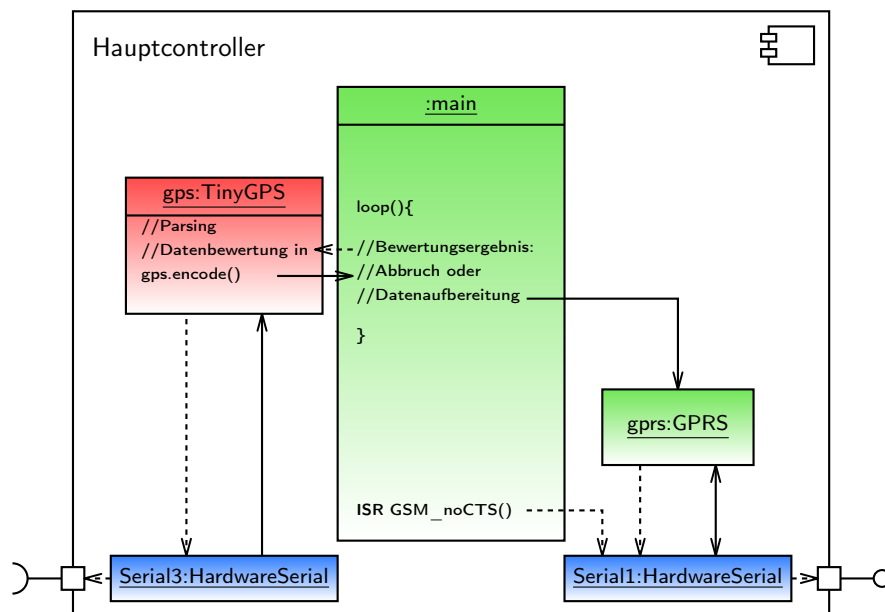


Abbildung 4.6.: Objekt- und Komponentendiagramm der Softwareimplementierung im Hauptcontroller

Grafik von links nach rechts abgearbeitet wird. Die jeweils zugehörigen Aufrufe sind der Übersichtlichkeit halber nicht dargestellt.

Die Abarbeitung der Programmschleife beginnt mit einem externen Kommunikationsvorgang, dem Einlesen von Daten vom GPS-Empfänger. Der Schnittstellenzugriff erfolgt über einen Treiberbaustein der Arduino-Umgebung (HardwareSerial, blau), welcher die Daten an den NMEA-Parser (TinyGPS, rot) übergibt. Der verwendete Parser abstrahiert den eintreffenden Datenstrom zu Paketen (NMEA-Sätzen) und führt auch deren Überprüfung durch (der erste Schritt gemäß 4.4). Ab Beginn der Überprüfung kann der Mikrocontroller keine weiteren eingehenden Nachrichten mehr verarbeiten, sie werden daher wie in Abb. 4.3 gezeigt verworfen.

Das Überprüfungsergebnis des Parsers (ermittelt in `gps.encode()`) stellt bereits die eigentliche Datenbewertung dar, wird jedoch erst nach dem Rücksprung in das Hauptprogramm (grün) ausgewertet. Dieses entscheidet anhand des Überprüfungsergebnisses, ob der Schleifendurchlauf mit gültigen Daten fortgesetzt oder aufgrund ungültiger Positionsdaten abgebrochen wird. Nur gültige Daten werden aufbereitet und über das GSM-Modem versandt (vgl. 4.4).

Die Datenaufbereitung besteht in ihrer bisherigen Form aus der Entgegennahme benötigter Einzelwerte vom Parser (Breitengrad, Längengrad, Zeitstempel, ...) sowie deren Konvertierung in für den Datenversand geeignete Formate. An dieser Stelle wäre eine Erweiterung der Software um zusätzliche Komponenten wie Filtersysteme zur Verminde- rung des Positionsrauschens theoretisch möglich, praktisch setzen die bereits vollständig ausgereizten Ressourcen des Controllers dem jedoch enge Grenzen.

Der Datenversand stellt wiederum einen externen Kommunikationsvorgang dar, er ist der bei weitem komplexeste und aufwändigste Teil der Software: Er nimmt sämtliche Module des absteigenden Astes aus Abb. 4.5 in Anspruch. Die aufbereiteten Daten werden zusammen mit weiteren Parametern (Zielserver, Port, ...) an den Modemtreiber (GPRS, grün) übergeben, welcher den dreistufigen Sendevorgang abwickelt. Zunächst erhält das Modem vom Treiber den Befehl, eine Verbindung zum Zielserver aufzubauen (vgl. Abb. 4.3, rechte Befehlskette und Abb. 4.4, Anforderung der Sendebereitschaft). Das Modem meldet den erfolgten Verbindungsaufbau (4.3, Verbindung aufgebaut bzw. 4.4, Sendebereitschaft hergestellt) und der Treiber signalisiert den Beginn der eigentlichen Übertragung, bevor er die Nutzdaten aus der Datenaufbereitung übergibt. Das Modem signalisiert wiederum Ende oder Fehler bei der Übertragung (nur 4.3) und mit dem Abbau der Verbindung endet die Kommunikation.

Die Länge dieses Vorganges und insbesondere die Menge der zu übermittelnden Daten können die Geräte auf beiden Seiten überfordern: Es drohen Pufferüberläufe und Datenverluste. Daher nutzt der Modemtreiber nicht nur die serielle Schnittstelle der Arduino-Umgebung (in Abb. 4.6 blau, rechts), sondern realisiert über zusätzliche Leitungen eine Flusskontrolle. Diese wird als neu implementierte Funktion im Abschnitt 4.6.2.3 detaillierter beschrieben.

4.6.2.2. Herkunft der Softwarebestandteile

Die Firmware des Mikrocontrollers besteht aus den zuvor skizzierten Modulen, welche aus verschiedenen Quellen stammen. Nicht modifiziert wurden dabei nur von der Arduino-Umgebung mitgebrachte Bestandteile, diese sind in Abb. 4.6 blau dargestellt. Komplette selbst geschaffene Module sind grün hervorgehoben, vor ihrer Verwendung modifizierte Bibliotheken Dritter rot. Die Abschnitte 4.6.2.3 und 4.6.2.4 gehen auf vorgenommene Veränderungen und Neuimplementierungen detaillierter ein.

4.6.2.3. Eigener Code

Vollständig selbst implementiert wurden das main-Objekt, die Flusskontrolle, der Watch-dog-Timer sowie – unter Verwendung des ursprünglichen Treibers als Vorlage – der Modemtreiber.

Das anonyme main-Objekt bildet den Kern eines jeden Mikrocontroller-Programms. Es handelt sich hierbei nicht um ein vollständiges Objekt im Sinne der objektorientierten Programmierung, sondern um ein logisches Element vergleichbar mit dem `init`-Prozess unixoider Betriebssysteme. Es beinhaltet u.a. den logischen Einsprungpunkt `main()`, welcher vereinfacht als Beginn der Programmausführung (der initiale Kontroll-Token) gesehen werden kann, sowie alle global gültigen Variablen, Konfigurationsparameter und die Wurzel aller Namensräume. Im Gegensatz zu Anwendungen in einem vollständigen Betriebssystem kehrt diese `main()`-Funktion jedoch niemals zurück, da die Software keinen Prozess des geordneten Herunterfahrens kennt.

Nach dem Einsprung in die `main()` beginnt zunächst eine Initialisierungsphase, in der die Peripherie des Controllers konfiguriert wird - in der Arduino-Umgebung erfolgt dies durch Aufruf der Funktion `setup()` (vgl. Datei `GPS_GSM_System.cpp`, elektronischer Anhang). Während dieser Initialisierungsphase werden in der vorliegenden Implementierung u.a. auch die seriellen Schnittstellen aktiviert, welche die Kommunikation zu den weiteren Komponenten ermöglichen, sowie die Setup-Prozeduren des Modemtreibers aufgerufen. Letztere enthalten sehr hardwarespezifische Startsequenzen, welche dazu dienen, das GSM-Modem in einen definierten Betriebszustand zu versetzen und die Verbindung zum Mobilfunknetz aufzubauen. Dieser Teil muss bei Änderungen der Hardware häufig angepasst werden.

Nach Abschluss der Initialisierungen geht der Controller in den Regelbetrieb über - eine Endlosschleife, welche bis zum Verlust der Energieversorgung oder zum Auslösen eines Resets ausgeführt wird. In der Arduino-Umgebung äußert sich dies in dem Rücksprung aus der Funktion `setup()` und dem folgenden Aufruf der Funktion `loop()`, welche auch in Abbildung 4.6 angedeutet ist. Diese Endlosschleife ist für die fortwährende Ausführung der im vorherigen Abschnitt beschriebenen Routinen zuständig und ist in ihrer Logik nicht von der konkreten Hardware abhängig. Die von dieser Funktion aus ausgerufenen Methoden können Teil anderer Objekte sein (echte Objekte i.S.d. OO). Die unmittelbar angesprungenen Funktionen dieser Objekte sind dabei i.d.R. ebenfalls noch vollkommen hardwareunabhängig, erst sie rufen ihrerseits tiefer angesiedelte Routinen auf, welche den Zugriff auf physische Komponenten realisieren und daher bei Änderungen am Gerät ggf. angepasst werden müssen. Im Idealfall ist dabei eine vollständige Trennung der verschiedenen Ebenen möglich – die TinyGPS-Klasse etwa arbeitet prinzipiell mit jedem beliebigen Objekt zusammen, welches ihr einen Strom von NMEA-Daten zur Verfügung stellen kann.

Die begrenzten Möglichkeiten der Controller erzwingen jedoch oft Zwischenlösungen, wie sie z.B. im Falle der Flusskontrolle des Modemtreibers realisiert wurden. Das Hauptprogramm arbeitet auch hier mit jedem beliebigen Treiber zusammen, welcher ein zum jetzigen identisches Interface bereitstellt, der Treiber selbst ist jedoch stark an die konkret verwendeten Komponenten angepasst und bietet keine weiteren Abstraktionsebenen mehr.

Bei dem Watchdog-Timer handelt es sich um eine zusätzliche Sicherungsschicht, welche in der Konzeption nicht vorkommt und die im Regelbetrieb auch nicht in Erscheinung tritt. Solche Timer sind weit verbreitet, sie stellen eine in den Controller integrierte Schaltung dar, welche dessen ordnungsgemäße Funktion überwacht: An mehreren Stellen des Quellcodes finden sich Aufrufe zum Zurücksetzen des Watchdog-Timers. Bleiben diese für mehr als 8 Sekunden aus, leitet die Schaltung einen vollständigen Reset des Mikrocontrollers ein, wodurch dieser auch die Neuinitialisierung sämtlicher externer Komponenten beginnt. Damit sollen unvorhersehbare Fehler wie das Kippen eines Bits im Controller-Speicher oder der Absturz der Modem-Software abgefangen werden, welche sonst den Weiterbetrieb des Ortungsmoduls unmöglich machen würden. Außerdem wird der Timer genutzt, um in bestimmten Situationen gezielte Resets auszulösen - etwa, wenn eine konfigurierbare Anzahl von Datenübertragungen zum Webservice in Folge fehlschlägt und somit die Möglichkeit besteht, dass eine Komponente des Systems in einen undefinierten Zustand geraten ist.

Der derzeitige Modemtreiber stellt nur eine temporäre Lösung dar: Er ersetzt den sehr fehlerbehafteten Treiber der GSM-Shield-Entwickler, ohne direkt Code von ihm zu verwenden. Die Sequenzen auszuführender AT-Befehle sowie zu setzender Einstellungen für bestimmte Aktionen wurden jedoch zu großen Teilen direkt von ihm übernommen, so dass es sich um eine Neuimplementierung, aber keine vollständige Neuentwicklung handelt. Der jetzige Treiber enthält lange Setup- und Hilfsroutinen, welche der Herstellung der Betriebsbereitschaft des Modems dienen, sowie höher angesiedelte Funktionen zur Unterhaltung von TCP/IP-Verbindungen und zum Stellen von HTTP-Anfragen. Es gibt Bestrebungen, Treiber für GSM-Modems innerhalb der Arduino-Umgebung zu standardisieren und über eine weitere Interface-Klasse zu abstrahieren, womit auch eine Trennung reiner Software-Funktionen (etwa HTTP-Implementierung) von hardwareunterstützten (TCP/IP) und hardwareabhängigen (z.B. Netzeinwahl) gegeben wäre. Diese begannen jedoch erst während der Arbeit an diesem Dokument, weswegen die derzeitige Code-Version ihnen noch keine Beachtung schenkt.

Die Flusskontrolle konnte nicht – wie in Abb. 4.5 konzipiert – als eigenes Modul implementiert werden, sondern besteht aus zwei separaten Teilen. Die Flusskontrolle vom Modem zum Hauptcontroller (bedeutet: Controller droht überlastet zu werden) ist Bestandteil des Modemtreibers. Dieser deaktiviert („legt“) das sog. *RTS*-Flag (Request to Send, eine physische Leitung), sobald sein Puffer sich füllt und aktiviert („setzt“) es erst wieder, wenn der Empfangspuffer geleert werden konnte. Bei deaktiviertem Flag unterbricht das Modem seine Sendungen.

Die Flusskontrolle vom Hauptcontroller zum Modem (bedeutet: Modem droht überlastet zu werden) konnte nicht auf diese Art realisiert werden, da die Senderoutinen des Arduino-Schnittstellentreibers (mangels Multitaskingfähigkeit des Betriebssystems) keine Abfrage des entsprechenden Flags während des Sendens zulassen. Stattdessen führt das *CTS*-Flag (Clear to Send, eine weitere physische Leitung neben RTS) auf einen Interrupt-Eingang des Mikrocontrollers: Legt das Modem das CTS-Flag, löst dies einen Interrupt im Hauptcontroller aus. Dieser Interrupt springt die ISR (Interrupt-Service-Routine) `GSM_noCTS()` im main-Objekt des Mikrocontrollers an, wodurch der Sendevorgang im Arduino-Schnittstellentreiber unterbrochen wird. Erst wenn das Modem seinerseits die Bereitschaft zum Empfang weiterer Daten signalisiert (das CTS-Flag setzt), kehrt die ISR zurück (zum Schnittstellentreiber) und der Sendevorgang kann fortgesetzt werden. Während der Übertragung einer einzigen Positionsinformation können diese Unterbrechungen in beide Richtungen durchaus mehrere Male auftreten – ein Tribut an die für solche Aufgaben sehr leistungsschwach dimensionierte Hardware.

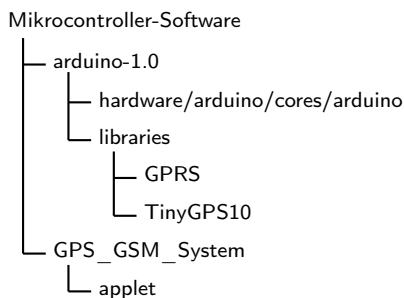
4.6.2.4. Modifizierung fremder Bibliotheken

Die TinyGPS-Bibliothek von Mikal Hart [40] wurde für ihren Einsatz im Ortungsmodul nur geringfügig modifiziert: Sie verwendet ursprünglich eine Klasse namens *NewSoftSerial* zur Implementierung einer seriellen Schnittstelle in Software. Da auf dem Arduino Mega 2560 insgesamt vier hardwareunterstützte Schnittstellen zur Verfügung stehen, konnte das GPS-Modul über eine solche angebunden werden. Die Unterstützung für die *NewSoftSerial* wurde daher aus der TinyGPS-Bibliothek entfernt und durch eine Hardwareanbindung ersetzt. Die Arduino-Umgebung bietet bislang keine Abstraktionsschicht, welche die minimalen Unterschiede in der Benutzung beider Schnittstellentypen abstrahiert, um derartige Modifikationen obsolet werden zu lassen.

4.6.3. Nutzung

4.6.3.1. Erstellung und Aufspielen auf den Controller

Für die Erstellung der Software werden vier Komponenten benötigt: Die Arduino-Umgebung, die modifizierte TinyGPS-Bibliothek, der Modemtreiber sowie der Code für das main-Objekt. Diese befinden sich gemeinsam im elektronischen Anhang dieser Arbeit. Es wird empfohlen, genau diese Komponenten zu verwenden, um Konflikte mit zwischenzeitlich möglicherweise aktualisierten Versionen zu vermeiden. Die Arduino-IDE selbst wird nicht zur Codeerzeugung verwendet, es werden lediglich zentrale Bestandteile des Betriebssystems compiliert und gelinkt. Die Verzeichnisstruktur der Software stellt sich wie folgt dar:



Das Verzeichnis `arduino-1.0` enthält die Arduino-Umgebung. Im Unterverzeichnis `hardware/arduino/cores/arduino` befinden sich zentrale Bestandteile des Arduino-Betriebssystems, darunter Initialisierungsprozeduren (vergleichbar mit dem Boot-Prozess normaler Rechner) sowie die im Rahmen dieser Arbeit genutzte Klasse `HardwareSerial`. Das Unterverzeichnis `libraries` enthält optionale Bibliotheken, welche auch aus anderer Quelle als vom Arduino-Projekt selbst stammen können. Hier sind der NMEA-Parser `TinyGPS` (Version 10) sowie der Modemtreiber `GPRS` angesiedelt.

Auf der Hauptebene befinden sich neben der Arduino-Umgebung weitere Verzeichnisse. Diese enthalten typischer Weise einzelne Programme, welche sich der zuvor genannten Bibliotheken und Komponenten bedienen. Es kann jeweils nur ein solches Programm übersetzt und auf den Mikrocontroller übertragen werden, es ist nicht möglich, mehrere Anwendungen gleichzeitig laufen zu lassen. Im Ordner `GPS_GSM_System` befindet sich der Quellcode, welcher das main-Objekt der Firmware des Ortungsmoduls bildet (genauer: dessen Routinen `loop()`, `setup()` und weitere, die eigentliche `main()` ist Bestandteil des Arduino-Kerns). Außerdem befindet sich dort ein `makefile`, welches den gesamten Build-Prozess automatisiert. Die in diesem `makefile` zu setzenden Einstellungen variieren für jedes Programm, da etwa der benötigte Funktionsumfang zu linkender Bibliotheken angegeben werden muss (z.B. `printf()` mit oder ohne Unterstützung für Gleitkommaarithmetik). Einige der vom `makefile` genutzten Tools benötigen absolute Pfadangaben, daher muss vor dem ersten Aufruf in Zeile 47 der Pfad zum (mitgelieferten) Arduino-Verzeichnis angepasst werden. Sollten die Installationspfade von Hilfsprogrammen von

den auf Unixoiden üblichen Pfadangaben abweichen, so müssen diese in Zeile 225ff. ebenfalls angepasst werden. Der Aufruf dieses `makefiles` und damit der Build-Vorgang der eigentlichen Mikrocontroller-Software erfolgt wie üblich durch die Angabe eines Targets - zur Auswahl stehen `make`, `clean` und `program`. Der Aufruf

```
$ make all
```

führt dementsprechend zum Start des Build-Prozesses. Zwischen- und Hilfsdateien sowie das Kompilat werden im Unterverzeichnis `applet` abgelegt, die Objektdaten (~.o) sämtlicher Klassen jedoch jeweils in dem Verzeichnis, in dem sich auch die Klasse selbst befindet. Alle Objekt-Dateien, welche unterhalb von `arduino-1.0` entstehen, werden gemeinsam zu einer statischen Bibliothek namens `core.a` gepackt. Das anonyme main-Objekt wird gegen diese Bibliothek gelinkt und das entstehende Kompilat als `main.hex` abgelegt. Diese Datei enthält den binären Befehlscode, welcher auf dem Mikrocontroller ausgeführt werden kann.

Unter Angabe des Targets `clean` werden all diese temporären Dateien wieder beseitigt. Das Target `program` versucht, das Kompilat unter Verwendung des im `makefile` eingestellten Programmieradapters auf den Mikrocontroller zu laden. Je nach der zur Verfügung stehenden Hardware müssen dafür die Zeilen 53ff. im `makefile` angepasst werden. Die Bootloader des Arduino-Projektes, welche das Nachladen neuer Software über die serielle Schnittstelle und damit ohne Programmieradapter erlauben, unterstützen seit der Version Arduino 1.0 den Einsatz von Watchdog-Timern nicht mehr. Sie können daher nicht verwendet werden.

4.6.3.2. Konfiguration

Die Firmware des Ortungsmoduls ist so ausgelegt, dass der Prototyp während seines Einsatzes nicht konfiguriert werden muss. Alle benötigten Einstellungen können vor dem Übersetzen einmalig vorgenommen werden und sind anschließend fest in der Software verankert. Nachfolgende Versionen des Gerätes sollen eine einfache Schnittstelle zur Konfiguration der wichtigsten Parameter bieten, welche dann auch in nichtflüchtigem, zur Laufzeit beschreibbarem Speicher (dem EEPROM des Controllers) vorgehalten werden. Für Nutzer des Ortungsmoduls relevante Parameter werden im Kopf der Datei `GPS_GSM_System.cpp` gesetzt, aus welcher das main-Objekt entsteht. Dies sind in Tabelle 4.1 aufgeschlüsselt.

Die Datei `GPRS.h` bietet darüber hinaus die Möglichkeit, wesentliche Parameter des Modemtreibers zu beeinflussen. Es wird empfohlen, diese auf ihren jetzigen Werten zu belassen bzw. nur im Rahmen der Weiterentwicklung der Hardware anzupassen. Tabelle 4.2 schlüsselt diese Parameter nach ihrer Bedeutung auf. Direkt unterhalb dieser Einstellungen können in der Datei darüber hinaus die globalen Debug-Ausgaben des Modemtreibers aktiviert bzw. deaktiviert sowie die zu verwendende Debug-Schnittstelle angepasst werden.

Tabelle 4.1.: Konfigurationsparameter des Ortungsmoduls

Parameter	Default-Wert	Bedeutung
GPRS_BAUD_RATE	115200L	Baudrate z. Kommunikation mit dem Modem
GPRS_INTERFACE	Serial1	Schnittstelle z. Kommunikation mit dem Modem
GPS_BAUD_RATE	115200L	Baudrate zur Kommunikation m. d. GPS-Empfänger
GPS	Serial3	Schnittstelle z. Kommunikation m. d. GPS-Empfänger
AIM_USERNAME AIM_PASSWORD	ortungsmodul_aim <entfernt>	Zugangsdaten zum AIM-Webservice
TRANSMISSION_FAIL_TRESHOLD	20	Anzahl fehlschlagender GPRS-Übertragungen, nach der ein Reset ausgelöst wird
GPS_data_stale	<Text> stale	Nachricht an den Webservice bei veralteten GPS-Daten
GPS_data_ok	<Text> ok	Nachricht an den Webservice bei aktuellen GPS-Daten

Tabelle 4.2.: Konfigurationsparameter des Modemtreibers

Parameter	Default-Wert	Bedeutung
GSM_LED	6	Pin Netlight-LED
GSM_RI	n	Pin Ring Indicator, ungenutzt
GSM_ON	8	Pin Modem-Einschaltsignal
GSM_RESET	9	Pin Modem-Resetsignal
GSM_CTS	2	Pin CTS-Leitung, Flusskontrolle
GSM_RTS	3	Pin RTS-Leitung, Flusskontrolle
GSM_DTR	n	Pin DTR-Leitung, ungenutzt
GSM_DCD	n	Pin DCD-leitung, ungenutzt
GPRS_REC_BUF_LEN	100	Größe Eingangspuffer
GPRS_CMD_BUF_LEN	280	Größe Kommandopuffer
GPRS_START_TIMEOUT	1000	Default-Timeout bis zum Beginn einer Antwort
GPRS_INTERCHAR_TIMEOUT	100	Default-Timeout zur Erkennung des Endes einer Antwort

5. Ergebnisse und deren Bewertung

5.1. Hardware

5.1.1. Derzeitiger Stand / erreichter Funktionsumfang

Der erste Prototyp besteht aus einzelnen, leicht trennbaren Modulen verschiedener Hersteller. Diese stellen alle für das konzipierte Gerät benötigten Funktionen zur Verfügung und arbeiten konfliktfrei zusammen. Die ursprünglichen Komponenten mussten teilweise modifiziert und um zusätzliche Baugruppen ergänzt werden, um den gestellten Anforderungen zu genügen. Die Ressourcen der Hardware sind mit dem jetzigen Entwicklungsstand weitestgehend erschöpft und die Möglichkeiten zur Steigerung der Leistung durch Veränderungen an den eingesetzten Baugruppen ausgereizt.

Trotz vorhandener Unzulänglichkeiten ist der erste Prototyp bereits funktionsfähig. Bei durchgeführten Testfahrten konnten Positionsdaten per GPS gewonnen und mittels GPRS an den Webservice übertragen werden, ohne dass nach dem Einschalten das Eingreifen eines Bedieners erforderlich gewesen wäre. Auf die Qualität dieser Daten geht Kapitel 5.3 noch einmal detaillierter ein.

5.1.2. Vorhandene Probleme

Die Hardware des Prototypen musste im Laufe der Entwicklung mehrfach und teilweise erheblich modifiziert werden, um aufgetretenen Problemen zu begegnen. So begünstigt die Führung einzelner Signale über lange Strecken sowie das qualitativ schlechte Layout einiger Leiterplatten das Auftreten kapazitiver Störungen, welche nicht vorhandene Signalepegel vortäuschen. Die zur Abhilfe integrierten Widerstände stellen nur ein Provisorium dar, welches erst durch eine Behebung der Ursachen abgelöst werden kann.

An den Einzelmodulen sind zahlreiche gegebenenfalls nutzbare Signale verfügbar, wurden aber nicht immer auch an die Verbinder zum Hauptcontroller herausgeführt. Die Vielzahl an Steckkontakten im Gesamtsystem provoziert schwer lokalisierbare Fehler durch zu hohe Übergangswiderstände oder mangelhafte Kontaktierungen. Um Doppelbelegungen zu

vermeiden, musste eine Reihe von Signalen von den seitens der Hersteller vorgesehenen Pins auf andere Eingänge des Hauptcontrollers umgeroutet werden, was weitere potentielle Fehlerquellen in Form von losen Verbindungen oder vertauschten Anschlüssen mit sich bringt.

Einzelne Ebenen des Prototypen erreichen durch ihre Bauhöhe beinahe die darüber liegenden Komponenten, was die Gefahr des Entstehens ungewollter elektrischer Verbindungen birgt und provisorische Isolationen erforderlich macht. Relevante Anschlüsse der Module sind zu verschiedenen Seiten und mit verschiedenen Bautiefen herausgeführt, was die Einpassung in einen gemeinsamen Baugruppenträger erschwert – das Fehlen eines schließbaren, geschirmten Gehäuses erhöht die Wahrscheinlichkeit ungewollter Beeinflussung durch umliegende elektronische Baugruppen sowie des Auftretens von ESD-Schäden.

Die Spannungsversorgung erfolgt über mehrere unabhängige, sehr ineffizient arbeitende thermische Regler, was die speisende Seite gerade während des Einschaltvorganges stark belastet und zu einem unnötig hohen Energieverbrauch des Gesamtsystems führt. Die damit einhergehende starke Erwärmung von Bauteilen in der Nähe der Energieversorgung verkürzt deren Lebensdauer erheblich und trägt zu einer vorschnellen Alterung des Gerätes bei.

5.1.3. Bewertung der Einsatztauglichkeit

Das Gerät ist als erstes in einer Serie von Prototypen noch nicht für den realen Einsatz ausgelegt. Bei Testläufen können die zuvor beschriebenen Probleme insbesondere mit den zahlreichen lösbaren Verbindungen im Gerät zum Auftreten von Fehlfunktionen führen. Der Mangel an Ausgabemöglichkeiten abseits der seriellen USB-Verbindung macht die ständige Anwesenheit eines Entwicklers erforderlich, der über ein externes Notebook die ordnungsgemäße Funktion des Gerätes überwacht. Wesentliche Bedienelemente wie der Reset-Taster sind nur schwer zugänglich und nicht klar als solche erkennbar. Die v.a. während der Entwicklungsphase notwendige Konfiguration von Parametern kann bislang nur vor der Übersetzung des Hauptprogramms erfolgen, was Änderungen während eines Einsatzes entsprechend erschwert.

5.2. Software

5.2.1. Derzeitiger Stand / erreichter Funktionsumfang

Die Software ist als Realisierung der Konzepte aus Kapitel 4.6.1 angelegt. Die Hauptaufgabe – das Einlesen von Positionsdaten und deren Weiterleitung an einen Webservice – wird erfüllt. Dazu wurden Module verschiedener Herkunft integriert und gegebenenfalls angepasst. Die für den späteren Realbetrieb relevanten Konfigurationsparameter sind bereits als solche vorgesehen und innerhalb des Controllers auch verfügbar, wenngleich derzeit noch nicht über Schnittstellen von außen beeinflussbar. Zur Sicherstellung der Funktion des Gerätes unter wechselnden Bedingungen sowie während längerer Einsatzzeiten wurde eine umfassende Fehler- und Ausnahmebehandlung integriert, welche auch eine Rückfallebene für den Eintritt unvorhergesehener Ereignisse bereithält.

Erweiterungen über den primären Einsatzzweck des Ortungsmoduls hinaus sind durch die limitierten Ressourcen der verwendeten Hardware nur sehr eingeschränkt möglich. Insbesondere Optionen für eine aufwändigere Datenverarbeitung vor dem Versand sind kaum vorhanden, wenngleich deren Integration seitens der Software sowohl möglich als auch explizit vorgesehen ist.

5.2.2. Vorhandene Probleme

Die Software enthält mit dem Modemtreiber ein Modul, welches ursprünglich von Dritten übernommen werden sollte, schlussendlich jedoch selbst geschrieben werden musste. Dieses ist sehr hardwarenah konzipiert und bietet nur den für die bisher realisierten Anwendungen minimal notwendigen Funktionsumfang. Eine durchaus mögliche Abstraktion verschiedener Ebenen dieses Treibers ist bislang nicht umgesetzt worden, für dessen Nutzung wichtige Bestandteile wie die Flusskontrolle sind nicht einmal vollständig mit ihm integriert. Die Beantwortung der Frage, ob dieser Treiber in zukünftigen Codeversionen weiterhin Verwendung finden oder durch eine generischere Lösung ersetzt werden wird, hat auch entsprechende Auswirkungen auf mit ihm interagierende Funktionen, die gegebenenfalls an ein neues Interface eines Ersatzmoduls angepasst werden müssen. Im Falle der Weiternutzung ist zudem eine umfassende Überarbeitung und Ausweitung der Möglichkeiten dieses Moduls notwendig.

5.2.3. Bewertung der Einsatzreife und Serientauglichkeit

Im derzeitigen Stand der Software sind notwendige Veränderungen für den Serieneinsatz bereits so weit wie möglich vorbereitet, viele Einstellungen sind aber noch für die Verwendung in einem Entwicklungsumfeld ausgelegt. So sind Parameter zur Demonstration des Konzeptes auf sehr sichere Werte festgelegt worden, schränken dadurch die Leistung des Gerätes jedoch ein (etwa Timeouts, Puffergrößen und Datenraten). Für einen Serieneinsatz müssen diese Beschränkungen minimiert werden, zudem werden Möglichkeiten zur Ausbringung von Updates und Konfigurationsänderungen an die in Nutzung befindlichen Geräte benötigt, welche wenig Personal binden und den notwendigen Einsatz kostenintensiver Hardware (z.B. Programmieradapter) minimieren. Die weitgehende Automatisierung des Build-Cycles ist hierbei nur ein erster Schritt im Aufbau entsprechender Möglichkeiten des Software- und Gerätemanagements. Auch das derzeitige Fehlen einer Implementierung zur Nutzung lokaler Datenspeicher wirkt sich hier aus, diese Punkte können und sollen aber mit der Einführung der nächsten Prototypengeneration behoben werden.

5.3. Funktion

5.3.1. Qualität und Aktualität übertragener Informationen

Die Qualität übertragener Informationen ist derzeit primär von der Leistung des GPS-Empfängers abhängig. Die Datenverarbeitung im Mikrocontroller nimmt keine Korrekturen der Werte vor, sondern filtert diese ausschließlich nach der Aussage *verwertbar* oder *nicht verwertbar*. Als verwertbar gelten dabei alle Positionsdaten, deren NMEA-Checksumme stimmt und die vom GPS-Modul selbst als valide einstuft werden, d.h. die unter vorhandenem Satellitenfix gewonnen wurden. Damit werden alle nach technischen Gesichtspunkten korrekten Positionsinformationen an den Webservice übermittelt. Die mit diesem Verfahren erzielten Genauigkeiten können prinzipiell Schwankungen von mehr als 50 m bis unter 1 m Abweichung von der realen Position aufweisen.

Das reale System erreicht auch unter suboptimalen Bedingungen typischer Weise Genauigkeiten im Bereich von unter 2 m, was für eine satellitenbasierte Ortung bereits sehr gut und für die Verfolgung der Bewegung eines Fahrzeuges mehr als hinreichend ist. Eine genauere Evaluierung der Ortungsleistung bietet Kapitel 3.3.5, die im elektronischen Anhang befindliche grafische Darstellung einer aufgezeichneten Testfahrt (**Visualisierung_Testfahrt_Ortungsmodul.png**) zeigt die Lage einzelner Messwerte gegenüber einer Karte der Umgebung auf (Kartenmaterial von Open Street Map). Weitergehende Anwendungen werden diese Daten i.d.R. nicht direkt verwenden, sondern je nach Anwendungszweck entsprechend aufbereiten. Gängige Verfahren sind hier insbesondere Map-Matching-Ansätze

zur exakten Abbildung gefahrener Strecken und Kalman-Filter zur Ergänzung oder Korrektur einzelner Messwerte, wofür die mit dem bestehenden System gewinnbaren Daten bereits eine sehr geeignete Grundlage bieten (siehe auch Kap. 2.2).

Die Aktualität der Daten meint die Zeitspanne, die vergeht, bis gewonnene Positionsdaten das System durchlaufen haben und in der Datenbank des Webservices verfügbar sind. Diese ist an sich nur für Echtzeitaufgaben interessant, da der GPS-Zeitstempel bereits jedem Wert einen exakten Entstehungszeitpunkt zuordnet, ist im real vorliegenden Ortungsmodul jedoch auch ein Maß für dessen Arbeitsgeschwindigkeit. Um so höher die Aktualität der Daten ist, desto kürzer wird auch der Abstand zwischen der Aufzeichnung zweier Positionen, was die Genauigkeit der daraus entstehenden Datenbasis verbessert.

Von der Erfassung einer Position bis zum Abschluss ihrer Übertragung an den Webservice benötigt das System in seiner derzeitigen Implementierung unter idealen Bedingungen etwa drei bis vier Sekunden. *Ideale Bedingungen* meint einen ausreichenden GPS-Empfang, um den Positionsfix halten zu können, sowie eine stabile GSM-Verbindung, welche keine oder nur vernachlässigbare Paketverluste zu verzeichnen hat. Dies ist auf freien Überlandstrecken und auf Autobahnen häufig der Fall. In städtischer Umgebung mit entsprechender Bebauung und weiteren Hindernissen (über die Straße ragende Bäume, Oberleitungen, Brücken und kurze Tunnel) steigen diese Verzögerungen auf typische 6 bis 8 Sekunden je Übertragung an, unterliegen dabei jedoch starken Schwankungen bis hin zum zeitweiligen Ausfall (siehe auch den nachfolgenden Abschnitt). Mit diesen Werten lassen sich zahlreiche Anwendungen wie sehr genaue Systeme zur Verspätungsvorhersage im ÖPNV durchaus realisieren, für Forschungsprojekte etwa zur Untersuchung des Schaltverhaltens von Ampeln oder zur Erfassung kurzfristig auftretender lokaler Verkehrshemmnisse ist aber eine noch aktuellere Datenbasis anzustreben.

5.3.2. Zuverlässigkeit des Gesamtsystems im Realeinsatz

Die Ortungsfunktion des Systems arbeitet auch unter erschwerten Bedingungen sehr zuverlässig. Die erfassten Positionswerte sind gegenüber den bestehenden Erwartungen hinreichend genau, nach unvermeidbaren Störungen wie Empfangsverlusten findet das System sehr schnell wieder zu seiner realen Position zurück (im Test jeweils unter 2 Sekunden). Die dadurch auftretenden Informationslücken umfassen i.d.R. nur einen oder zwei Messpunkte, was durch anwendungsspezifische Verfahren vor der Nutzung der Daten wieder ausgeglichen werden kann (etwa Kalmanfilter zur Schätzung fehlender Werte oder Map-Matching-Techniken zur Restaurierung der gefahrenen Route; siehe auch Kap. 2.2.4).

Die Datenübermittlung via GPRS erweist sich nach den bisherigen Testergebnissen als sehr störanfällig. Selbst einfache Hindernisse nahe an oder über der Strecke führen zu

einer rapiden Verminderung der Übertragungsleistung bis hin zu sporadischen Totalverlusten der Verbindung und somit zu einer Erhöhung der Übertragungszeit je Messwert, was sowohl die Aktualität der Datenbasis gefährdet als auch die Anzahl gewonnener Messpunkte drastisch reduziert. Diese Empfindlichkeit gegenüber der Umwelt wurde von einer GSM-basierten Übertragung nicht erwartet. Sie ist Gegenstand weiterhin laufender Untersuchungen zu den möglichen Ursachen, wobei zu vielen wichtigen Faktoren wie der Netzabdeckung in den Testgebieten, der Provider- bzw. Frequenzabhängigkeit der Störungen sowie dem verwendeten Übertragungsprofil zwischen Modem und Basisstation noch keine belastbaren Daten vorliegen.

Die Bewegungsgeschwindigkeit des Fahrzeuges hatte entgegen anfänglicher Befürchtungen keinen messbaren Einfluss auf das System, weder auf die Ortungs- noch die Übertragungsleistung. Getestet wurde auf freier Strecke bis ca. 200 km/h, was zumindest den im ÖPNV auftretenden Geschwindigkeitsbereich hinreichend abdeckt.

6. Zusammenfassung

Ziel der vorliegenden Arbeit war die Entwicklung eines kostengünstigen, universell einsetzbaren Systems, welches seine per GPS erfassten Positionsdaten kontinuierlich und in Echtzeit mittels GSM an einen zentralen Webservice übertragen kann.

Dazu wurden Anforderungen an das zu entwickelnde System definiert, Grundlagen der dahinter stehenden Technologie recherchiert und ein Konzept zur Entwicklung des Gerätes ausgearbeitet. Mittels entsprechender Auswahl- und Evaluationsprozesse wurden Entscheidungen für zu verwendende Einzelkomponenten der Hardware getroffen und ein erster Prototyp entwickelt. Parallel zu dessen Optimierung entstand über mehrere Fragmente und Teststufen hinweg ein Softwarekonzept, welches in der Implementierung der derzeit finalen Firmwareversion mündete.

Mit der Erprobung des geschaffenen Systems gelang der Nachweis der Funktionsfähigkeit des konzipierten Gerätes. Es konnten Positionsdaten per GPS erfasst und mittels GSM/GPRS zyklisch im Abstand weniger Sekunden an einen bereitgestellten Webservice übertragen werden. Die bisherige Evaluation lieferte wertvolle neue Erkenntnisse zur Leistungsfähigkeit und zu noch bestehenden Schwachstellen der konkreten Umsetzung, welche in die weitere Entwicklung des Systems einfließen werden.

Die Fortsetzung der Arbeiten an diesem Projekt ist fest geplant und hat bereits begonnen. Anstehende Meilensteine wie die Schaffung einer zweiten Prototypengeneration und die damit einhergehende Überarbeitung der Software sind definiert. Die vorliegende Arbeit schließt damit ein erstes, erfolgreiches Kapitel dieses Projektes ab – und möchte Grundlage sein für das, was folgt.

7. Ausblick

7.1. Laufende und geplante Weiterentwicklung des Gerätes

7.1.1. Hardware

7.1.1.1. Ablösung des Ebenen-Systems

Die kontinuierliche Weiterentwicklung des Ortungsmoduls zur Verwendung im Rahmen des AIM-Projektes ist seitens des DLR fest geplant. Mit dem erbrachten Beweis der Funktionsfähigkeit steht daher im nächsten Schritt eine dringend notwendige Bereinigung der Hardware an. Zahlreiche der unter 5.1.2 genannten Probleme lassen sich durch einen geeigneten Neuaufbau beheben, Unzulänglichkeiten maßgeblich verbessern.

Die erste Evolutionsstufe nach dem im Rahmen dieser Arbeit vorgestellten Prototypen soll jedoch nicht nur Mängel und Fehler bereinigen, sondern auch die Umgestaltung der Hardware zur Serienproduktion vorbereiten. Die bisher provisorisch ausgeführten Modifikationen sollen dabei integraler Bestandteil des nächsten Systems werden. Zur Professionalisierung des Geräteaufbaus ist die Reduzierung der Ebenen sowie die Erhöhung der Packungsdichte elektronischer Komponenten anzustreben, die Vereinheitlichung des Zuganges zu wichtigen Schnittstellen sowie die Verbesserung sekundärer Entwurfsmerkmale (Energieversorgung, Pegelwandler, Wartungsmöglichkeiten).

Zur Erreichung dieser Ziele soll eine Basisplatine entworfen werden, welche beinahe alle benötigten Komponenten selbst trägt. Die Umsetzung dieser Platine als Schaltplan und als entsprechendes Layout bietet die Gelegenheit, die Zahl potentiell fehlerträchtiger Verbindungen zu minimieren und das Routing entscheidend zu verbessern. Die Bestückung soll weitestgehend mit platz- und energiesparenden SMD-Bauteilen erfolgen, aufgrund der niedrigen Stückzahl von ein bis drei Exemplaren jedoch manuell durchgeführt werden. Das verwendete GSM-Modul SIMCOM 900 sowie der GPS-Empfänger SkyTraq Venus 634FLPx sind ohne entsprechende maschinelle Ausstattung kaum verlötbar, weswegen das Sparkfun-GPS-Modul und das Tochterboard des GSM-Shields als einzige Komponenten zunächst weiterhin über Stecksockel mit der Hauptplatine gekoppelt werden sollen.

Eine komplette Neukonzeption der derzeit für jedes Modul separat aufgebauten Energieversorgung kann die Effizienz des Gesamtsystems erheblich steigern und die Belastung für die Einspeisung wie auch für interne Baugruppen minimieren helfen. Durch eine Absenkung der Hauptspannung auf einen für alle Komponenten nutzbaren Pegel unterhalb von 4 V ist es möglich, diverse Pegelwandler einzusparen und das bisherige System mehrerer thermischer Festspannungsregler durch einen einzelnen, effizienteren Schaltregler mit einem Weitbereichseingang von mindestens 10 ... 28 V zu ersetzen.

Dieser Entwurf wird zur Beseitigung auftretender Errata sowie zur Fertigstellung der Software bis zum Erreichen der Betriebsfähigkeit dienen. Gegebenenfalls werden mehrere aufeinander folgende Hardwareversionen produziert, um das Design weiter zu optimieren oder nicht ideal arbeitende Komponenten zu ersetzen.

7.1.1.2. Erreichen der Serienreife

Eine vorläufig letzte Version des Gerätes soll in einer Kleinserie von 20 bis 200 Exemplaren produziert und im Rahmen des AIM-Projektes eingesetzt werden. Für deren Nutzung ist es wichtig, diese in standardisierte Gehäuse einsetzen zu können, welche in industrielle Geräteträger passen und die Verwendung anpassbarer Frontplatten erlauben. Dafür sollen sämtliche externe Schnittstellen an einer oder zwei Seiten des Gerätes platziert und auf eine einheitliche Tiefe ausgerichtet werden. Zusätzliche Elemente wie Status-LEDs und bei Bedarf einfache Eingabetaster helfen die praktische Einsatztauglichkeit zu steigern.

Die Kosten- und Budgetsituation wird über die genaue Stückzahl sowie die Art der Fertigung entscheiden, was Auswirkungen auf die mögliche technische Gestaltung hat. Sollte eine maschinelle Fertigung der Kleinserie möglich sein, können auch die Hochfrequenzkomponenten (GPS-Empfänger und Modem) mit in die Hauptplatine integriert werden. Die für eine solche automatisierte Bestückung notwendigen Losgrößen sind vor Beginn der Planungen zu klären und die Resultate in den Entwurf der dritten Generation mit einzubeziehen.

Mit dem erfolgreichen Einsatz dieser Exemplare gilt die Entwicklung als vorerst abgeschlossen und das Design steht für die Weiternutzung im Rahmen anderer Projekte zur Verfügung.

7.1.2. Software

Neben der Anpassung an kommende Hardwaregenerationen wird die unmittelbare Weiterentwicklung der Software auf die Umsetzung neuer Vorhaben abzielen. Dies betrifft Basisfunktionalitäten wie die Integration von Status-LEDs ebenso wie die Bereinigung der Treibersituation oder die Ergänzung höherwertiger Funktionen, welche die bisherigen Möglichkeiten des Ortungsmoduls in entscheidenden Punkten erweitern.

Eine dieser Erweiterungen soll lokalen Speicher, etwa den EEPROM des Controllers oder eingesteckte microSD-Karten, zur Ablage von Konfigurationsparametern und log-Dateien nutzbar machen. Dies ist eine notwendige Grundlage für den Feldeinsatz der Ortungsmodule, da damit für einfache Veränderungen nicht mehr das gesamte Betriebssystem neu geladen werden muss und vorbereitete Änderungen gegebenenfalls auch vom Wartungspersonal der die Geräte tragenden Fahrzeuge eingespielt werden können. Inwieweit ein Ausbau der Datenverarbeitung im Ortungsmodul notwendig wird, hängt von den Ergebnissen der Evaluierungsphase mit dem jetzigen sowie dem nachfolgenden Prototypen ab.

7.2. Anwendungsmöglichkeiten der nun verfügbaren Informationen

Das Ortungsmodul dient im Rahmen der Anwendungsplattform Intelligente Mobilität zur Datenerfassung von Fahrzeugen öffentlicher Verkehrsträger. Sein primärer Zweck ist die Bereitstellung von Daten für wissenschaftliche Zwecke.

Diese ermöglichen Auswertungen und Beobachtungen zur uhrzeit-, tages-, wochen- und saisonabhängigen Verkehrsdichte und -lage im Beobachtungsgebiet mit einer Präzision bis zu einzelnen Straßen und Kreuzungspunkten. Die Erkennung von Engstellen, Unfallschwerpunkten und zeitabhängigen Phänomenen bildet unter anderem die Grundlage zur verbesserten Modellierung realer Phänomene im Rechner und ermöglicht die Evaluation neuartiger Ansätze zur Verkehrsbeeinflussung zuerst im exakten Simulationsmodell. Für die praktischen Erprobung eröffnet die tagesaktuelle Datenbasis die Gelegenheit, auftretende Einzelereignisse wie auch gezielt durchgeführte Maßnahmen in ihren Auswirkungen sofort zu beobachten und zu bewerten.

Ferner bietet sich mit der direkten Beobachtung der Verkehrslage aus der Sicht einzelner *Verkehrsteilnehmer* (anstatt der Draufsicht von Verkehrskameras, Fahrzeugsensoren u.a.) die Chance, bereits existierende Anwendungen auf ihre Wirksamkeit hin zu überprüfen und diese gegebenenfalls zu verbessern – dazu gehören Grüne-Welle-Schaltungen

und für ÖPNV und Einsatzdienste automatisch oder per Funkanforderung umschaltende Lichtsignalanlagen ebenso wie existierende Systeme zur Verspätungsvorhersage und Kundeninformation an Haltestellen des Nahverkehrs.

Zukunftsperspektiven sehen darüber hinaus Anwendungsmöglichkeiten für das Modul unter anderem im Individualverkehr: Die Beobachtung etwa der kompletten Fahrzeugflotte eines lokalen Dienstleisters (Taxiunternehmen, Speditionen, Kurierdienste) lässt Vergleiche zwischen verschiedenen Fahrzeugen in ähnlichen oder identischen Situationen zu, womit z.B. eine Beurteilung verschiedener Strategien zur Routenplanung und Navigation und damit letztendlich auch eine Optimierung dieser Systeme möglich wird. Inwieweit dabei auftretende Synergieeffekte mit aktuellen Entwicklungen im Fahrzeugbereich, etwa automatischen Notrufsystemen und integrierten Möglichkeiten zum Flottenmanagement, genutzt werden können, bleibt abzuwarten.

Abbildungsverzeichnis

2.1. Beispiel zu Möglichkeiten der Ortung in Abhängigkeit von den verfügbaren Datenquellen	6
2.2. Die Krümmung des Kurvenabschnittes ergibt sich aus dem Winkel zweier sich schneidender Geraden (Tangenten) sowie der zwischen ihnen liegenden Strecke.	13
2.3. Die nicht approximierten Krümmungswerte einer Kurve als Funktion der Bogenlänge. Bildquelle: [58, S. 17]	13
2.4. Richtungswinkel einer Kurve als Funktion der Bogenlänge, vor und nach der Glättung. Bildquelle: [58, S. 26]	14
2.5. Komponentendiagramm des Ortungsmoduls	28
3.1. Übersicht der aufgezeichneten Messpunkte	41
3.2. Detailaufnahme des Bereiches um den Referenzpunkt	42
3.3. Bewegungsspur des MOD-GPS	43
3.4. Bewegungsspur des Venus-GPS	44
3.5. Bewegungsspur des Javad-Delta-Empfängers	44
3.6. Komponenten des ersten Prototypen im Endzustand	49
3.7. Das Ortungsmodul im Endzustand. Das Schnittbild zeigt die Elektronik in der unteren Gehäuseschale, ganz links die Antenne für GSM, daneben GPS.	51
4.1. Entwicklungsprozess eingebetteter Systeme als Aktivitätsdiagramm	53
4.2. Interaktionsübersichtsdiagramm der Komponenten des Ortungsmoduls	61
4.3. Sequenzdiagramm zum Kommunikationsablauf innerhalb des Ortungsmoduls	63
4.4. Aktivitätsdiagramm zum normalen Betriebsablauf im Hauptcontroller	64
4.5. Objekt- und Komponentendiagramm zur Modularisierung und Abstraktion der Software im Hauptcontroller	65
4.6. Objekt- und Komponentendiagramm der Softwareimplementierung im Hauptcontroller	66

Tabellenverzeichnis

3.1. Vergleich von Arduino Uno und Arduino Mega 2560	31
3.2. Parameter der verglichenen GPS-Module lt. Datenblättern [38] und [39] . .	38
3.3. Mittelwerte der gemessenen Startzeiten	39
3.4. Mittelwerte der gemessenen Positionen und deren Abweichung vom Referenzpunkt	40
4.1. Konfigurationsparameter des Ortungsmoduls	73
4.2. Konfigurationsparameter des Modemtreibers	73
B.1. Messung der Kaltstartzeit	B-2
B.2. Messung der Warmstartzeit	B-2
B.3. Messung der Heißstartzeit	B-2

Quellenverzeichnis

- [1] http://de.wikipedia.org/wiki/Decca-Navigationssystem#Fahrzeugverfolgung_an_Land, letzter Zugriff: 22.01.2012
- [2] <http://de.wikipedia.org/wiki/GPS#Geschichte>, letzter Zugriff: 22.01.2012
- [3] Vor 50 Jahren – HOCHBAHN entwickelt weltweit erstes Busüberwachungssystem; Webartikel, 19. Januar 2012
<http://www.itcs-info.de/neuigkeiten/488-vor-50-jahren-hochbahn-entwickelt-weltweit-erstes-busueberwachungssystem.html>, letzter Zugriff: 22.01.2012
- [4] http://de.wikipedia.org/wiki/Digital_Mobile_Radio#Einordnung_gegen_C3_BCber_anderen_Funkstandards, letzter Zugriff: 22.01.2012
- [5] <http://www.etsi.org/WebSite/Technologies/DigitalMobileRadio.aspx>, letzter Zugriff: 22.01.2012
- [6] *ETSI (Hrsg.):* Digital Mobile Radio (DMR); Broschüre, Frankreich
<http://www.etsi.org/WebSite/document/Technologies/LEAFLETS/DigitalMobilRadio.pdf> und Anhang, letzter Zugriff: 22.01.2012
- [7] *Sven-Olaf Suhl:* DMR: Digitaler und analoger Betriebsfunk mit demselben Endgerät; Webartikel, heise mobil, 18.04.2007
<http://www.heise.de/mobil/meldung/DMR-Digitaler-und-analoger-Betriebsfunk-mit-demselben-Endgeraet-168784.html>, letzter Zugriff: 22.01.2012
- [8] <http://en.wikipedia.org/wiki/DPMR>, letzter Zugriff: 22.01.2012
- [9] *ETSI (Hrsg.):* Digital Private Mobile Radio (dPMR); Broschüre, Frankreich
http://www.etsi.org/WebSite/document/Technologies/LEAFLETS/digitalPrivateMobileRadio_UK.pdf und Anhang, letzter Zugriff: 23.01.2012
- [10] <http://www.dpmr-mou.org>, letzter Zugriff: 22.01.2012
- [11] <http://www.dpmr-mou.org/dpmr-faq.htm>, letzter Zugriff: 22.01.2012
- [12] <http://de.wikipedia.org/wiki/B%C3%BCndelfunk#Einsatzbereiche>, letzter Zugriff: 23.01.2012
- [13] http://de.wikipedia.org/wiki/Terrestrial_Trunked_Radio#Technik, letzter Zugriff: 23.01.2012

- [14] *Jürgen Kuri*: Irritationen nach TV-Sendung über den digitalen Behörden-Funk; Webartikel, heise News, 15.10.2008
<http://www.heise.de/newsticker/meldung/Irritationen-nach-TV-Sendung-ueber-den-digitalen-Behoerden-Funk-211331.html>, letzter Zugriff: 23.01.2012
- [15] *Bundesverband Professioneller Mobilfunk e.V.*: Motorola Solutions erweitert TETRA-Netze von Vattenfall Europe Distribution; Webartikel, Köln / Idstein, 22. November 2011
<http://www.pmev.de/aktuell/120-motorola-solutions-erweitert-tetra-netze-von-vattenfall-europe-distribution>, letzter Zugriff: 23.01.2012
- [16] *Sven-Olaf Suhl, Volker Briegleb*: Flughafen Leipzig/Halle erhält TETRA-Digitalfunknetz; Webartikel, heise Netze, 24.04.2007
<http://www.heise.de/netze/meldung/Flughafen-Leipzig-Halle-erhaelt-TETRA-Digitalfunknetz-170783.html>, letzter Zugriff: 23.01.2012
- [17] *Sven-Olaf Suhl*: Flughafen Köln-Bonn modernisiert sein Tetra-Digitalfunknetz; Webartikel, heise Netze, 07.08.2007
<http://www.heise.de/netze/meldung/Flughafen-Koeln-Bonn-modernisiert-sein-Tetra-Digitalfunknetz-160457.html>, letzter Zugriff: 23.01.2012
- [18] *I.G.T. Informationsgesellschaft Technik mbH (Hrsg.)*: Koordinierung des operativen Flughafenbetriebs; Fachartikel, I.G.T. Informationsgesellschaft Technik mbH, München, 07.03.2011
<http://www.sicherheit.info/si/cms.nsf/si.ArticlesByDocID/1116700?Open>, letzter Zugriff: 23.01.2012
- [19] *Ministeriums für Wirtschaft, Energie, Bauen, Wohnen und Verkehr des Landes Nordrhein-Westfalen (Hrsg.)*: Digitaler Funk/Regiofunk; 16.09.2009 <http://www.fachportal.nahverkehr.nrw.de/index.php?id=92>, letzter Zugriff: 23.01.2012
- [20] <http://de.wikipedia.org/wiki/TETRAPOL>, letzter Zugriff: 23.01.2012
- [21] http://www.cassidian.com/en_US/web/guest/communication_network_security, letzter Zugriff: 23.01.2012
- [22] http://tetrapol.com/community/tetrapol_forum.html, letzter Zugriff: 23.01.2012
- [23] <http://tetrapol.com/community/supporters.html>, letzter Zugriff: 23.01.2012
- [24] http://www.cassidian.com/en_US/web/guest/tetrapol-communication-system2, letzter Zugriff: 23.01.2012
- [25] http://www.cassidian.com/en_US/web/guest/tetra-communication-system2, letzter Zugriff: 23.01.2012
- [26] Glossar, heise online
<http://www.heise.de/glossar/entry/Tetrapol-397263.html>, letzter Zugriff: 23.01.2012

- [27] *Jens Christiansen*: Tetrapol
<http://www.digitaler-bos-funk.de/tetrapol/tetrapol.htm>, letzter Zugriff: 23.01.2012
- [28] http://www.tetrapol.com/technology/why_tetrapol.html, letzter Zugriff: 23.01.2012
- [29] *Detlef Borchers, Jürgen Kuri*: Digitaler Behördenfunk: Funknetzkonvergenz in Aachen; Webartikel, heise News, 23.08.2005
<http://www.heise.de/newsticker/meldung/Digitaler-Behoerdenfunk-Funknetzkonvergenz-in-Aachen-124966.html>, letzter Zugriff: 23.01.2012
- [30] *IT-AmtBw LSC PrÖA*: TETRAPOL Bw; Bundesamt für Informationsmanagement und Informationstechnik der Bundeswehr, Koblenz, 22.11.10
<http://www.it-ambw.de/portal/a/itamtbw/vorhab/tetrap>, letzter Zugriff: 23.01.2012
- [31] *Sven-Olaf Suhl*: Digitalfunk: Bundeswehr ordert Tetrapol-Systeme bei EADS; Webartikel, heise News, 16.09.2005
<http://www.heise.de/newsticker/meldung/Digitalfunk-Bundeswehr-ordert-Tetrapol-Systeme-bei-EADS-130648.html>, letzter Zugriff: 23.01.2012
- [32] Digitalfunksysteme (TETRAPOL / TETRA); Poly-Verlag, Bassersdorf/Schweiz, 27. April 2011
<http://www.poly-verlag.ch/digitalfunk-tetra-tetrapol.htm>, letzter Zugriff: 23.01.2012
- [33] *Bundesamt für Bevölkerungsschutz BABS*: POLYCOM – Sicherheitsnetz Funk der Schweiz; 29.07.2009
<http://www.bevoelkerungsschutz.admin.ch/internet/bs/de/home/themen/polycom.html>, letzter Zugriff: 23.01.2012
- [34] *Bundesamt für Kommunikation BAKOM*: TETRAPOL Faktenblatt; 13.06.2006
Download unter <http://www.bakom.admin.ch/themen/technologie/01220/index.html> und Anhang, letzter Zugriff: 23.01.2012
- [35] *Klaus Bäumer*: EMVU und Technik, Newsletter 4/2004 (Auszug)
http://www.fgf.de/publikationen/newsletter/einzeln/NL_04-04/DigitalerBuendel1funk_04-04d.pdf und Anhang, letzter Zugriff: 23.01.2012
- [36] <http://arduino.cc/en/>, letzter Zugriff: 26.01.2012
- [37] <http://arduino.cc/en/Main/Boards>, letzter Zugriff: 26.01.2012
- [38] Datenblatt SkyTraq Venus 634FLPx
http://www.sparkfun.com/datasheets/GPS/Modules/Skytraq-Venus634FLPx_DS_v051.pdf und Anhang, letzter Zugriff: 27.01.2012

- [39] Datenblatt Olimex MOD-GPS
<http://www.olimex.com/dev/pdf/OTHER/MOD-GPS.pdf> und Anhang, letzter Zugriff: 27.01.2012
- [40] TinyGPS-Library von Mikal Hart
<http://arduiniiana.org/libraries/tinygps/>, letzter Zugriff: 30.01.2012
- [41] Arduino GSM Shield von Open Electronics – Beschreibung
<http://www.open-electronics.org/arduino-gsm-shield/>, letzter Zugriff: 30.01.2012
- [42] Arduino GSM Shield von Open Electronics – Arduino-Library
<http://code.google.com/p/gsm-shield-arduino/downloads/list>, letzter Zugriff: 30.01.2012
- [43] http://de.wikipedia.org/wiki/Hidden_Markov_Model, letzter Zugriff: 06.02.2012
- [44] <http://de.wikipedia.org/wiki/Markow-Kette>, letzter Zugriff: 06.02.2012

Literaturverzeichnis

- [45] *M. S. Grewal, L. R. Weill, A. P. Andrews*: Global Positioning Systems, Inertial Navigation, and Integration, 2. Auflage; John Wiley & Sons; New Jersey, 2007
- [46] *I. Colomina*: Towards Intelligent Systems for Integrated Sensor Orientation; Paper; in: Integrated Sensor Orientation, Colomina/Navarro (Hrsg.); Wichmann; Heidelberg, 1995
- [47] *J. Wendel*: Integrierte Navigationssysteme, 2. Auflage; Oldenbourg Verlag; München, 2011
- [48] *Y. Zhao*: Vehicle Location and Navigation Systems; Artech House; Norwood, 1997
- [49] *M. Witt (Hrsg.)*: GPRS – Start in die mobile Zukunft; MITP-Verlag; Bonn, 2000
- [50] *E. Schnieder (Hrsg.)*: Verkehrsleittechnik – Automatisierung des Straßen- und Schienenverkehrs; Springer-Verlag; Berlin, Heidelberg, 2007
- [51] *C. Scholz*: Fahrzeugautarkes Ortungssystem für Lokomotiven auf der Basis schlupfbehalteter Wegmessung; Fortschritt-Berichte VDI, Reihe 12, Nr. 559; VDI-Verlag; Düsseldorf, 2003
- [52] *B. Walke*: Mobilfunknetze und ihre Protokolle 2, 3. Auflage; Teubner-Verlag; Stuttgart, Leipzig, Wiesbaden, 2001
- [53] *E. Lertes*: Funkortung und Funknavigation; Vieweg; Braunschweig, Wiesbaden, 1995
- [54] *W. Mansfeld*: Satellitenortung und Navigation, 3. Auflage; Vieweg+Teubner; Wiesbaden, 2010
- [55] *T. Lefebvre, H. Bruyninckx, J. De Schutter*: Nonlinear Kalman Filtering for Force-Controlled Robot Tasks; Springer Tracts in Advanced Robotics, Volume 19; Springer-Verlag; Berlin, Heidelberg, 2005
- [56] *E. Neuburger*: Einführung in die Theorie des linearen Optimalfilters; R. Oldenbourg Verlag; München, Wien, 1972
- [57] *K.-W. Schrick (Hrsg.)*: Anwendungen der Kalman-Filter-Technik: Anleitungen und Beispiele; Methoden der Regelungstechnik; R. Oldenbourg Verlag; München, Wien, 1977
- [58] *R. Czommer*: Leistungsfähigkeit fahrzeugautonomer Ortungsverfahren auf der Basis von Map-Matching-Techniken; Dissertation; Verlag der Bayerischen Akademie der Wissenschaften; München, 2001

- [59] *Deutsches Zentrum für Luft- und Raumfahrt e.V. (Hrsg.):* Entwicklung eines Demonstrators für Ortungsaufgaben mit Sicherheitsverantwortung im Schienengüterverkehr; Berichte aus dem DLR-Institut für Verkehrssystemtechnik, Band 8; Braunschweig, 2009
- [60] *S. Thrun, W. Burgard, D. Fox:* Probabilistic Robotics; The MIT Press; Cambridge, London, 2006

Abkürzungsverzeichnis

AGPS	Assisted GPS
BOS	Behörden und Organisationen mit Sicherheitsaufgaben
CDMA	Code Division Multiple Access
DGPS	Differential GPS
DMR	Digital Mobile Radio
dPMR	digital Private Mobile Radio
EGNOS	European Geostationary Navigation Overlay Service
ETSI	European Telecommunications Standards Institute
GNSS	Global Navigation Satellite System
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GSM-R	GSM Railway
HMM	Hidden Markov Model
IC	Integrated Circuit
INS	Inertiales Navigationssystem
ITCS	Intermodal Transport Control System
ÖPNV	Öffentlicher Personennahverkehr
RBL	Rechnergestütztes Betriebsleitsystem
TETRA	Terrestrial Trunked Radio
TTFF	Time to first Fix

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit bisher bei keiner anderen Prüfungsbehörde eingereicht, sie selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Georg Richter, Wernigerode,
den 13. Februar 2012

Anhang

A. Vergleich verschiedener Hardwarekomponenten

Das folgende Dokument listet eine Reihe verfügbarer Hardwarekomponenten auf, die zur Verwendung im Ortungsmodul in Frage kämen. Die Komponenten sind nach Einsatzzweck gegliedert und die für den jeweiligen Zweck relevanten Parameter zum Vergleich aufgelistet. Ferner angegeben ist der jeweils günstigste Preis sowie der Link zum günstigsten und – so verfügbar – weiteren Anbietern der Komponente gemäß dem Stand zur Recherche im November 2011.

Wo jeweils mehrere zusammengehörige Artikel nur im Verbund als eine Komponente eingesetzt werden können, sind diese entsprechend gruppiert und der Gesamtpreis für eine solche Gruppe mit angegeben. Da sich die Breite einer Tabelle jeweils über mehrere Seiten erstreckt, sei auf die im Anhang befindliche elektronische Version des Dokumentes verwiesen (Datei `Vergleich_Hardwareoptionen.ods`).

Controllerboards				
Name	Controller	Takt (MHz)	Spannung	USB-versorgbar?
Arduino Uno	ATMega328	16	5 V	j
Arduino Mega 2560	ATMega2560	16	5 V	j
Arduino Uno Ethernet	ATMega328		intern 3,3 und 5 V	n
USB-Adapter für Uno Ethernet (FTDI Friend)	wie Uno, benötigt zur Programmierung aber FTDI Friend			-
Ethernet Pro	ATMega328	16	intern 3,3 und 5 V	n
FTDI basic breakout			5 V (auf 3,3 V umschaltbar)	
WIZNet Embedded Web Server Module	ATMega 128			n

SD Logging Shields		
Name	Für	Beinhaltet
microSD Shield	Uno (Mega?)	µSD-Sockel und Prototypenfläche

Prototyping-Shields		
Name	Für	Beinhaltet
ARDUINO/PROTO	Uno	Leiterplatte, leer
ARDUINO/MEP	Mega2560	Leiterplatte, leer
Arduino ProtoShield Kit	beide	Leiterplatte mit Komponenten (Pin Header, LEDs, B...

Ethernet-Shields				
Name	Chip	für	Anbindung	Ethernet
Arduino Ethernet Shield	Wiznet W5100	Uno/Mega	SPI	1x

USB-UARTS	weitere UARTs	I ² C (TWI)	SPI	Ethernet	SD/MMC	GSM	GPS	Preis
1	0	1	2 (1 durch USART, durch USB belegt)	n	n	n	n	22,5 € - 2
1	3	1	5 (4 durch USART, teilw. belegt)	n	n	n	n	43,70 €
0	1	1	2 (1 durch USART, belegt)	j	j	n	n	43,70 €
-	-							15,50 €
0	1	1	2 (1 durch USART, belegt)	Wiznet W5100	n	n	n	45,80 €
-	-							11,50 €
0	2	1	1	Wiznet W5300	n	n	n	46,00 €

Preis	Anbieter	Link
14,00 €	TinkerSoup	http://www.tinkersoup.de/product_info.php?products_id=243

Preis	Anbieter	Link
5,90 €	Elmicro	http://elmicro.com/de/arduino-shields.html
6,90 €	Elmicro	http://elmicro.com/de/arduino-shields.html
14,00 €	TinkerSoup	http://www.tinkersoup.de/product_info.php?products_id=356

SD/MMC	Preis	Anbieter	Link
microSD	32,50 € - 33,00 €	Elmicro / TinkerSoup	http://elmicro.com/de/arduino-shields.html
			http://www.tinkersoup.de

Anbieter	Link									
Elmicro / TinkerSoup	http://elmicro.com/de/arduino.html		http://www.tinkersoup.de/product_info.php?products_id=385					http://arduino.cc/en/Main/ArduinoEthernetShield		
Elmicro	http://elmicro.com/de/arduino.html									
Elmicro	http://elmicro.com/de/arduino.html		http://www.adafruit.com/products/418					http://www.arduino.cc/en/Main/ArduinoEthernetShield		
Elmicro	http://elmicro.com/de/arduino.html									
TinkerSoup	http://www.tinkersoup.de/product_info.php?products_id=623									
TinkerSoup	http://www.tinkersoup.de/product_info.php?products_id=322									
TinkerSoup	http://www.tinkersoup.de/product_info.php?products_id=247									

rdduinoBoardUno	
ain/ArduinoBoardEthernet	

ain/ArduinoBoardEthernet	

GPS-Shields	Beschreibung oder			
Name	Chip	Spannung	Update-Frequenz:	Anbindung
Lösung 1:				
Arduino XBee Shield mit:				
GPS Bee	u-blox 5 NEO-5Q	2,7 - 3,6 V	4 Hz	UART
S3ED/GPSANT	Mini-Patchantenne, 10cm kabellänge			
SFE09145	Adapterkabel U.FL auf SMA, Länge ca. 10cm			
SFE00177	GPS-Patch-Antenne mit, ca. 12cm Kabellänge			
SFE00464	GPS Magnetfuß-Antenne (gekapselt), ca. 5m Anschlussleitung			
Gesamtkosten:				
Lösung 2a:				
Adafruit GPS Shield mit einem der folgenden Module:				
USGlobalSat EM-406A	SiRF StarIII	5 - 6,5 V	1 Hz	USART 4800
Trimble Lassen iQ	Trimble	3,3 V		USART 2400-19200
Lösung 3: (nicht bekannt, ob unten gelistete Module mit einem der GPS Shields zusammenarbeiten)				
TinkerSoup / Sparkfun GPS Shield mit einem EM-406, EM-408, EB-85A oder einem der folgenden Module:			// enthält einen Konnektor für eine Back	
Locosys LS20031	MediaTek MT3318	3,3 V	5 Hz	USART 57600
Micro Modular Technologies MN5010HS "Micro-Mini"	SiRF StarIII	3,3 - 5,5 V	1 Hz	
Venus GPS Board	SkyTraq Venus634FLPx	3,3 V	10 Hz	UART (TTL), SPI
GS407	u-blox 5H	3,3 V	4 Hz	UART (UBX-Protoc
Parallax PX28146	SONY CXD2951GA-4 (Engine PMB-248)	5 V		einzelner Pin, Baug
Polstar PMB-648	SiRF StarIII	3,3 - 5 V		UART 4800 (TTL/R
Polstar PMB-688	SiRF StarIII	3,3 - 5 V		UART 4800 (TTL or
PX28502	GPS Magnetfuß-Antenne (mit Gehäuse), Anschlussleitung 2,70m			
Olimex Mod-GPS	SiRF StarIII	3,3 - 6 V		USART 19200 (CM

GSM-Shields		Beschreibung oder		
Name	Chip	Spannung (V)	Beherrscht	
Arduino GSM Shield	SIMCom SIM 900	3,9 V / Konverter in Board	GSM/GPRS	
Cellular Shield	SM5100B	3,8 V / Konverter im Board	GSM/GPRS, TCP/IP	

	ext. Antenne?	SD/MMC	Preis	Anbieter	Link					
		n	17,50 €	Elmicro	http://elmicro.com/de/xbee.html					
	U.FL-Buchse		42,90 €	Elmicro	http://elmicro.com/files/seedstudio/gps_bee-man					
	<i>U.FL-Stercker</i>		6,90 €	Elmicro	http://elmicro.com/de/gpsbee.html					
			3,80 €	Elmicro	http://elmicro.com/de/gpsbee.html					
	<i>SMA-Stecker</i>		10,10 €	Elmicro	http://elmicro.com/de/gpsbee.html					
	<i>SMA-Stecker</i>		9,70 €	Elmicro	http://elmicro.com/de/gpsbee.html					
	Gesamtkosten:		67,30 - 74,30 €							
		SD	19,50 \$	Adafruit	http://www.adafruit.com/prod	http://www.ladyada.net/make/gps				
	n, Keramikantenne integriert		60\$ / 58,30€	Adafruit / El	http://www.a	http://elmicro.co	http://www.ladyada.net/make/gps			
)	H.FL-RF		57\$+19\$ (An	Sparkfun	http://www.s	http://www.spar	http://www.ladyada.net/make/gps			
up-Batterie		n	14,50 €	TinkerSoup	http://www.tinkersoup.de/product_info.php?products_id=229					
	n, Keramikantenne integrie	n	47,60 €	Elmicro	Verlötung erforderlich!					
	n, Keramikantenne integrie	n	67,50 €	Elmicro	http://elmicro.com/de/gps.html					
	SMA-Stecker	n	37,50 €	Elmicro	http://elmicro.com/de/gps.html					
ol)	n, Helixanenne integriert	n	75,00 €	Elmicro	http://elmicro.com/de/gps.html					
ruppe enthält eigenen F	n, Keramikantenne integrie	n	60,60 €	Elmicro	http://elmicro.com/de/gps.html					
S232)	n, Keramikantenne integrie	n	30,30 €	Elmicro	http://elmicro.com/de/gps.html					
ly)	Keramikantenne integriert u	n	34,70 €	Elmicro	http://elmicro.com/de/gps.html					
	<i>MMCX-Konnektor</i>	n	10,10 €	Elmicro	http://elmicro.com/de/gps.html					
OS)	Gesamtes Modul ist in geka	n	52,75 €	Elmicro	http://elmicro.com/de/gps.html					

[illegible]

[illegible][illegible]

Gehäuse				
Produkt	Abmaße	Ausgelegt für		
Arduino/Case	-	Arduino (Uno oder Mega) und Ethernet Shield		
Box for Arduino	87*64*28mm	Arduino (Uno oder Mega), Ethernet Shield, Batterie		
Reichelt o.ä., da obige zu teuer und zu klein				

IMU-Shields	
Name	Beschreibung
Lösung 1:	
ArduPilot Mega	Eigentlich ein Autopilot für UAVs, enthält einen ATmega2560, taugt als Basiscontroller
Lösung 2:	
ArduIMU+ V2	Einfache IMU mit ATmega328, könnte als Basiscontroller dienen
	GPS-Modul für ARDUIMU+ V2, GS-407 U-Blox5, Helixantenne
Lösung 3:	
9 DOF Razor IMU	Enthält einen ATmega328 zur Vorverarbeitung, serielle Ausgabe 38400 Baud, 3.5 - 16V Eingang
Lösung 4:	
ArduIMU Sensor Board	6 DOF, davon 4 auf diesem Board, zwei weitere auf Tochterboards (nicht verfügbar)

LCD-Shields		
Name	Beschreibung	LCD
Lösung 1:		
Critical Velocity LCD Shield Kit	Basisplatine mit Piezo für beliebige HD44780-kompatible LCDs	-
	LCD 16*2 mit Hintergrundbeleuchtung	16*2 / BL
	LCD 16*4 ohne Hintergrundbeleuchtung	16*4
	LCD 16*4 mit Hintergrundbeleuchtung	16*4 / BL
weitere Lösungen:		
Color LCD Shield	graph. Farbdisplay (mit BL) + 3 frei nutzbare Buttons, SPI-Anbindung	Nokia 6100 Color LCD
OLED Display 1.5" Shield for Arduino	OLED-Farbdisplay (ohne BL) + 5-Positionen-Joystick, USART oder 4DGL	1,5" uOLED-128-G7
OLED Display 1.7" Shield for Arduino	OLED-Farbdisplay (ohne BL) + 5-Positionen-Joystick, USART oder 4DGL	1,7" uOLED-160-G7
anderes Display (eBay-Variante?), da alle zu teuer		

B. Messprotokolle zum Vergleich ausgewählter GPS-Module

B.1. Anmerkungen

Die im Folgenden protokollierten Messungen dienen zur Ermittlung der jeweiligen Startzeiten der zu vergleichenden Module aus verschiedenen Situationen heraus (Kalt-, Warm- und Heißstart) sowie der Positions-Updaterate. Die Module befanden sich jeweils auf freiem Feld ca. 1 m über Grund und wurden in ihrer Sicht zum Himmel nicht behindert. Alle Messungen fanden an derselben Position statt.

Konfigurationssoftware Venus-GPS: SkyTraq 0.4.604

Konfigurationssoftware MOD-GPS: SiRFDemo 3.87

B.2. Updateraten

Im elektronischen Anhang dieser Arbeit findet sich die Ausgabe der GPS-Module während der Ermittlung der Updaterate. Aus ihnen wurde anhand der NMEA-Zeitstempel jeweils ein 120 Sekunden langes Zeitfenster ausgewählt, um die Anzahl der darin auftretenden Positionsupdates zu ermitteln. Gezählt wurden die aufgetretenen GPGGA-Sätze.

	Venus-GPS	MOD-GPS
Datum:	27.01.2012	27.01.2012
Uhrzeit:	ab 20:22 MEZ	ab 20:07 MEZ
Konfig. Updaterate:	10 Hz	1 Hz
Anz. erwarteter Sätze:	1200	120
Anz. aufgetretener Sätze:	1173*	120

* Es traten mehrfach kurze Übertragungsfehler auf, welche die Differenz erklären. Diese sind dem störanfälligen Versuchsaufbau zuzuschreiben, in dem der Arduino als Relais zwischen Empfänger und Computer agieren muss.

B.3. Startzeiten

Tabelle B.1.: Messung der Kaltstartzeit

Messung, Datum, Uhrzeit	Venus-GPS	MOD-GPS
1: 25.01.2012 16:35	29 Sekunden	35 Sekunden
2: 25.01.2012 16:42	31 Sekunden	37 Sekunden
3: 25.01.2012 16:46	32 Sekunden	36 Sekunden
Mittelwert:	30,7 Sekunden	36,0 Sekunden

Tabelle B.2.: Messung der Warmstartzeit

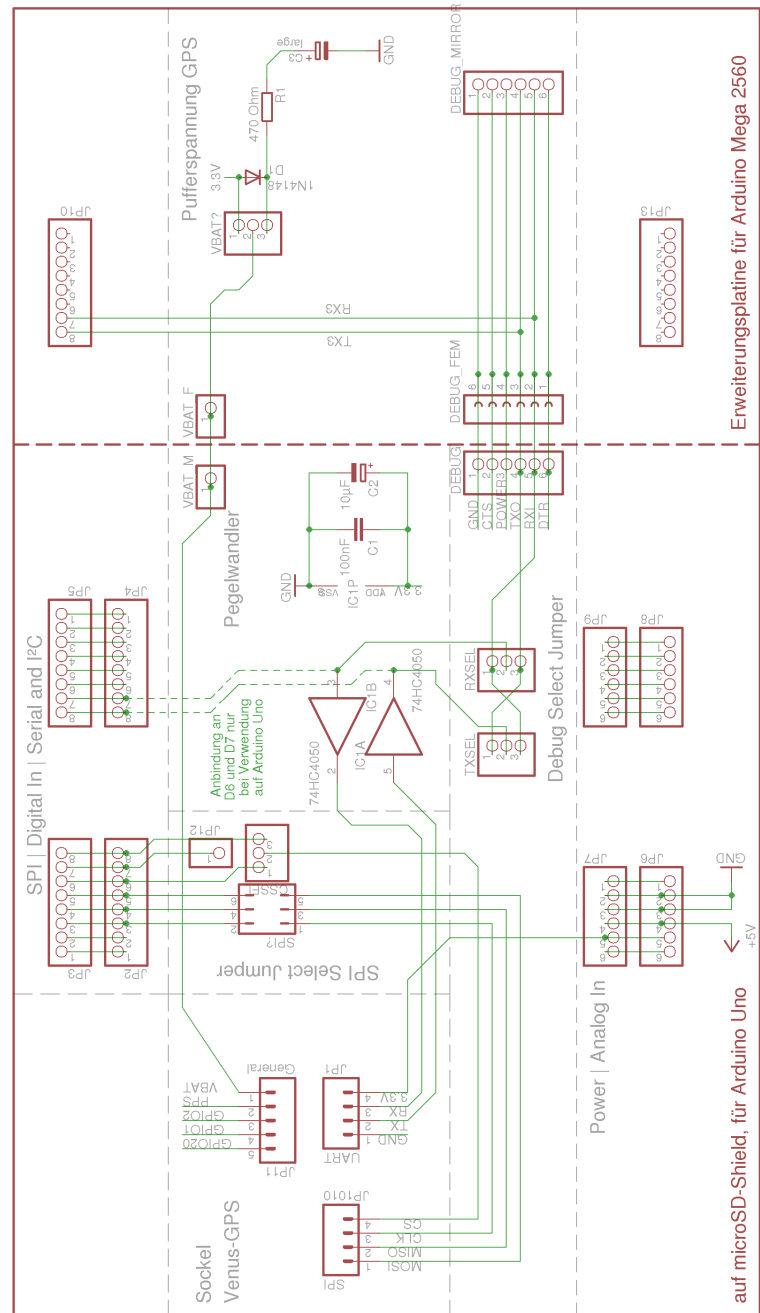
Messung, Datum, Uhrzeit	Venus-GPS	MOD-GPS
1: 26.01.2012 10:17	5 Sekunden	36 Sekunden
2: 26.01.2012 21:33	8 Sekunden	22 Sekunden
3: 27.01.2012 17:17	7 Sekunden	33 Sekunden
Mittelwert:	6,7 Sekunden	30,3 Sekunden

Tabelle B.3.: Messung der Heißstartzeit

Messung, Datum, Uhrzeit	Venus-GPS	MOD-GPS
1: 26.01.2012 10:20	< 1 Sekunde	< 1 Sekunde
2: 26.01.2012 21:35	< 1 Sekunde	< 1 Sekunde
3: 27.01.2012 17:22	< 1 Sekunde	< 1 Sekunde
Mittelwert:	< 1 Sekunde	< 1 Sekunde

C. Schaltpläne

C.1. Pegelanpassung des Venus-GPS-Moduls



D. Sourcecodes und technische Unterlagen

Die nachfolgende CD enthält die im Rahmen der Arbeit erstellten Unterlagen und Quelltexte.